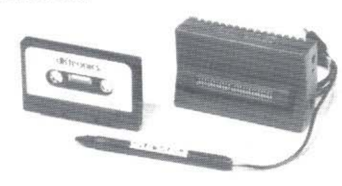


5,– DM
ÖS 43,–
sfr 5,–
CPC
Schneider
INTERNATIONAL
9
September 1985
CPC 6128
DAS KRAFTPAKET
DER CPC MIT 128K
Programme
SYMBOL EDITOR
LINE RUNNER
ZEHNKAMPF TEIL II
Hardware
SPEICHERERWEITERUNG
NEUE 5 1/4"-FLOPPY
LIGHTPEN
Tools
WERKZEUGE FÜR
MASCHINENSPRACHEPROGRAMMIERER
JETZT 8 SEITEN MEHR!!

dk'tronics Lightpen für CPC 464

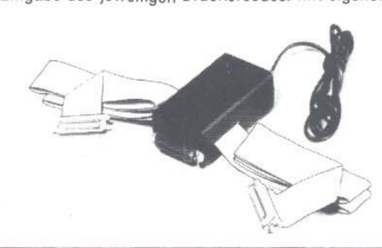
Das Werkzeug für den Grafikprogrammierer: Pixelweise Auflösung, volle Ausnutzung der Farb- und Grafikmöglichkeiten des CPC, auch für Grünmonitor, umfangreiches Hilfsprogramm mit Menuesteuerung über Pictogramme, dadurch kinderleichte Handhabung, kompatibel zu DDI-1 und MP1.
Komplett mit Handbuch und Hardcopyroutine.



Best.-Nr. 6400
nur 99,– DM

NEU 6510 Druckerinterface von Data Media für Schneider CPC 464/664

2 verschiedene Drucker können wahlweise betrieben werden. Steuerung über die Software, durch Eingabe des jeweiligen Druckercodes. Mit eigener Stromversorgung!



198,– DM

Staubschutzhauben

aus weichem Kunstleder, beigefarben für Computer und Peripherie.

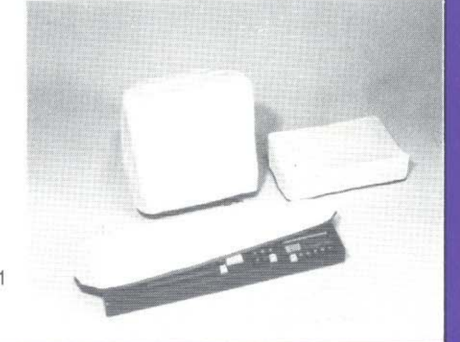
f. Grünmonitor
Best.-Nr. 6409
37,50 DM

f. Farbmonitor
Best.-Nr. 6410
37,50 DM

für Drucker NLQ 401
Best.-Nr. 6408
22,50 DM

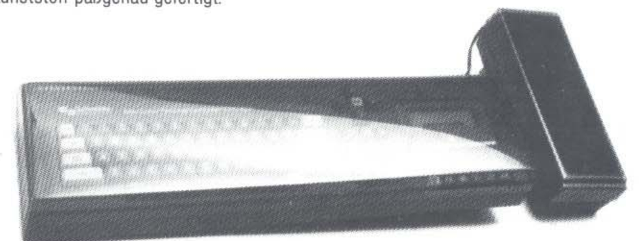
f. CPC 464
Best.-Nr. 6406
22,50 DM

f. Floppy DDI-1
Best.-Nr. 6407
19,90 DM



DATA MEDIA Abdeckhauben

Formschön und stabil! Schützen Sie Ihre wertvolle Hardware vor Staub, Schmutz und Beschädigung. Die Abdeckhauben sind aus durchsichtigem, rauchglasfarbenem Hartkunststoff paßgenau gefertigt.

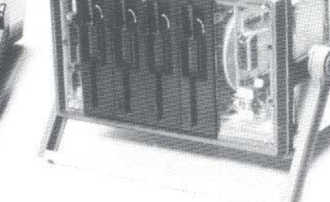


Best.-Nr. 6119 Abdeckhaube für CPC 464 Computer **35,– DM**
Best.–Nr. 6120 Abdeckhaube für Floppy DDI-1 **29,– DM**

5 1/4"-Laufwerk von Data Media NEU

Das professionelle Speichermedium für Ihren CPC 464!
Einsatzmöglichkeit als eigenständiges Laufwerk unter CP/M. Maximale Speicherkapazität von 3.2 MB bei 4 Laufwerken, eingebauter Controller sowie ein Maschinensprache-Monitor auf Eprom.





Zum Lieferumfang gehören:
- Laufwerke mit Netzteil
- Controller-Karte mit Eprom
- je 1 deutsches Handbuch für CP/M und Bedienungsanleitung des Monitors

Lieferbar ab 10.09.85

Folgende Konfigurationen sind möglich:

Best.-Nr.	Anz. d. Laufwerke	Speicherkapaz.	Preis
6130	2	1.6 MB	1598,– DM
6140	3	2.4 MB	2298,– DM
6150	4	3.2 MB	2998,– DM

Speichererweiterung von Data Media für CPC 464 NEU

Lieferbar ab 10.09.85

Bestell-Nr. 6170	64KB	Speichererweiterung	**198,– DM**
Bestell-Nr. 6171	128KB	Speichererweiterung	**298,– DM**
Bestell-Nr. 6172	256KB	Speichererweiterung	**398,– DM**
Bestell-Nr. 6173	512KB	Speichererweiterung	**498.– DM**

Der Preishit

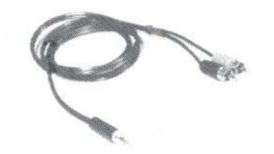
Speichererweiterung 64 KB, nicht aufrüstbar **nur 128,– DM**

Leerdisketten

3" (10er Pack) **129,00 DM**

Zubehör von John Hall:

Druckerkabel für Schneider-Drucker, 150 cm lang
Best.-Nr. 6402 **DM 79,– DM**



Joystick-Adapter
Zum Anschluß von 2 Standard-Joysticks an Schneider Computer **17,90 DM**
Best.-Nr. 6404

Schneider-Stereo-Kabel
3,5 mm Klinken-Stecker/
2 Chinch Stecker
150 cm lang
Best.-Nr. 6400 **16,50 DM**

Ohne Abbildung:
Schneider-Monitor-Verlängerungskabel,(2 Kabel) für CPC Monitor
Best.-Nr. 6403 **19,50 DM**

Ohne Abbildung:
Schneider-Stereo-Kabel
3,5 mm Klinken-Stecker/5 pol. DIN-Stecker, 150 cm lang
Best.-Nr. 6401 **16,50 DM**

The Stick

Völlig neues Steuergefühl durch Einhandbedienung!
Durch die Bewegung der Hand besonders geeignet für schnelle Action.
Mittels Saugfuß auch stellbar.
Fabrikat: John Hall
Best-Nr. 6405 **49,– DM**

Wie bestellen?
Die Bestellkarte im Innenteil der CPC heraustrennen, ausfüllen und an DATA MEDIA senden. Lieferung erfolgt umgehend! Zahlung per Vorkasse oder Nachnahme zzgl. Porto-bzw. Nachnahmegebühr. (Nachnahme ins Ausland ist nicht möglich.)

data media gmbh MAILORDER



Weitere Artikel in unserem Gesamtkatalog. Bitte anfordern (2,– DM für Rückporto beilegen.)
DATA MEDIA GmbH, Bereich Mailorder, Postfach 1263, 4620 Castrop-Rauxel
Telefonische Auskünfte: 0231/12 50 71
In der Schweiz: Data Media Schweiz, Bungertstraße 15, 8802 Zürich-Kilchberg, Tel.: (01) 715 36 19

Impressum

Schneider CPC International
erscheint im Data Media GmbH Verlag

Chefredakteur
Christian Widuch (verantwortlich)

Redaktion
Stefan Ritter (SR), Thomas Morgen (TM)

Gestaltung
Renate Wells, Christine Mayer

Grafik/Illustration
Heinrich Stiller

Fotografie
Gerd Köberich

Anzeigenleitung
Wolfgang Schnell
Es gilt die Anzeigenpreisliste Nr. 1 vom 1.1.1985

Freie Mitarbeiter
Horst Franke (HF), Jens Barthe,
Harold Eggink, Thomas Mischke,
Hans Matschat, Ernst Hügli, Alwin Ertl,
Gerd Schröder, Joachim und Christan Hofer

Anschrift Verlag/Redaktion
Data Media GmbH
-Bereich Verlag-
Postfach 250
3440 Eschwege
Telefon: 05651/8702

Vertrieb
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Vertrieb Österreich
Pressegroßvertrieb
Salzburg Ges.mbH & Co. KG
Niederalm 300
5081 Anif
Tel.: 06246/3721

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
"Schneider CPC International" erscheint monatlich am Ende des Vormonats.
Einzelpreis DM 5,-/sfr. 5,-/ÖS 43,-

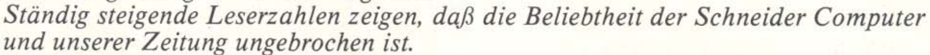
Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland:
Jahresbezugspreis: DM 60,-
Halbjahresbezugspreis: DM 30,-
Europäisches Ausland:
Jahresbezugspreis: DM 90,-
Halbjahresbezugspreis: DM 45,-
Außereuropäisches Ausland:
Jahresbezugspreis: DM 120,-
Halbjahresbezugspreis: DM 60,-
Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim Verlag schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel.
Das Abonnement verlängert sich automatisch um 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.
Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung.
Honorare nach Vereinbarung (die Zustimmung zum Abdruck wird vorausgesetzt).

Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.



Liebe Leser

Im Autoradio, auf der Fahrt von Frankfurt nach Kassel, hörte ich kürzlich einen ironischen Bericht über das berühmte Sommerloch, das vielen Branchen Kopfzerbrechen bereitet. Nachdem dieser erste Sommer für die CPC nun endgültig vorüber ist, kann ich nur feststellen, daß dieses Sommerloch zu uns offensichtlich nicht vorgedrungen ist! Im Gegenteil:



Ständig steigende Leserzahlen zeigen, daß die Beliebtheit der Schneider Computer und unserer Zeitung ungebrochen ist.

Da in der Zwischenzeit mit der Neuvorstellung des 128K Computers „Schneider CPC 6128" bereits drei Geräte dieses Herstellers auf dem Markt sind, haben wir uns daher entschlossen, den Umfang unserer Zeitschrift ab dieser Ausgabe um 8 Seiten zu erhöhen. Selbst jetzt haben wir noch Probleme, alle interessanten Aspekte rund um den Schneider in die vorhandenen Seiten unterzubringen. 8 Seiten mehr bedeuten mehr Informationen, mehr Computer-Spaß.

Am 30. August öffnet die Internationale Funkausstellung in Berlin ihre Pforten. Bereits vor 2 Jahren, auf der vorherigen Funkausstellung, hatte die Computerei neben den traditionellen Produkten der Unterhaltungselektronik einen wesentlichen Anteil am Gesamtumfang dieser Messe. In diesem Jahr wird dies noch stärker der Fall sein. Wie bereits in Dortmund und in Köln, so stellen wir uns auch in Berlin unseren Lesern. Die Computerhalle 9A, Stand Nr. 9A 14 ist unsere Messeadresse. Wir laden alle unsere Leser zu einem Besuch an unserem Stand ein. Der Meinungsaustausch zwischen Lesern und dem Verlag ist eine wertvolle Gelegenheit, dieses Medium optimal auf die Bedürfnisse der User einzustellen.

Noch einmal möchte ich mich für die Mitarbeit all der Leser bedanken, die an unserer Fragebogenaktion im Heft 6 mitgewirkt haben. Ca. 3000 Zusendungen waren mehr als wir erwartet hatten! Die Gewinner des mit dieser Aktion verbundenen Preisausschreibens sind in diesem Heft auf Seite 59 veröffentlicht; die Gewinne werden in Kürze zugestellt.

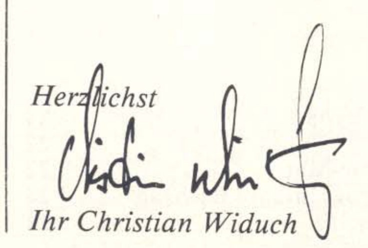
Mit unserer Databox haben wir offensichtlich einen weiteren Haupttreffer bei unseren Lesern gelandet. Obwohl die Resonanz auf diesen neuen Service unsere Erwartungen bei weitem übertraf und zudem in die Urlaubszeit fiel, gelang es uns dennoch, die Auslieferung der Databox ohne Verzögerungen vorzunehmen. Hierzu zwei grundsätzliche Bemerkungen auf die Fragen vieler Leser:

1. *Aus organisatorischen Gründen bieten wir diesen Service vorerst nicht als Abonnement an.*
2. *Databox ist vorläufig nur als Kassettenversion erhältlich.*

Auf Anregung vieler Leser arbeiten wir an einer Version der Databox für die zurückliegenden Hefte. Wir werden in der Ausgabe 10 weiter über dieses Thema informieren.

Einen angenehmen Computerherbst wünscht Ihnen Ihre CPC-Redaktion.

Herzlichst



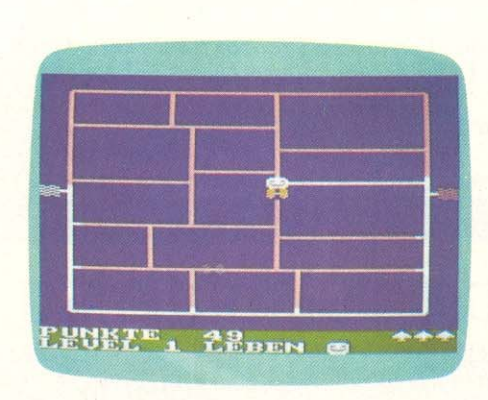
Ihr Christian Widuch



Als Brückenwärter hat man es schwer. Testen Sie Ihre Qualifikation mit dem Spiel „Bridge It"



Unsere Erfahrungen mit dem CPC 664 möchten wir Ihnen nicht vorenthalten. In unserem Bericht stellen wir Ihnen nicht nur die neuen Basic Befehle vor, sondern geben auch Tips zur Soft- und Hardware des 664.



Line-Runner ist ein Action- und Geschicklichkeitsspiel in Anlehnung an professionelle Spielhallen-Hits. Lassen Sie sich diesen Leckerbissen nicht entgehen!

Berichte

Serie

Programme

Lehrgänge

Fortsetzung der beliebten Kurse

Software Reviews

Hardware

Tips und Tricks

Referenzkarte

Rubriken

Das Warten hat sich gelohnt:

hier ist sie nun, die TURBO-PASCAL-SCHNEIDER-VERSION! Mit deutschem Handbuch und auf der Schneider-Diskette. Wenn Sie also einen CPC 464 besitzen und das Diskettenlaufwerk dazu, können Sie ab sofort sowohl das Turbo-Pascal in seiner einfachen Ausführung als auch die erweiterte Version bestellen.

Mit dieser speziell angepaßten Version haben alle Besitzer eines CPC 464 Zugang zur meistverkauften Programmiersprache mit weltweit über 300.000 Kopien. Turbo-Pascal ist damit der Quasi-Industrie-Standard und nicht nur einer der billigsten Pascal-Compiler, sondern auch deren schnellster.

In Turbo-Pascal ist ein Editor enthalten, der sehr viele Features des berühmten Wordstar bietet. Mit diesem integrierten Editor, der Lokalisierung von Fehlern im Programmtext, sowie der erstaunlichen Kompaktheit und Geschwindigkeit ist Turbo-Pascal das erste Maschinencode-»Voll-Pascal« mit echter Anfänger-Eignung.*

In der erweiterten Version bekommen Sie einen Riesensprachumfang für die Grafik (Turtle-Grafik) und den Soundgenerator. Turbo 3.0 beinhaltet eine Reihe von Assembler-Routinen, die Sie bei der Kompilierung in Ihre Turbo-Programme einbinden können. Sie bietet Ihnen die erweiterten Grafik-Kommandos, die im Folgenden beschrieben werden:

TURBO 3.0 für den Schneider CPC

- **Turtle-Grafik**
  Die Turbo-Turtle-Grafik basiert auf dem »turtle«-Konzept und erlaubt mit seiner durchs Bild marschierenden Schildkröte (turtle) auch denen einen leichten Umgang mit der Grafik, die sich mit anderen Grafiksystemen schwer anfreunden konnten. Mit einfachen Algorithmen lassen sich interessante Bilder gestalten, indem man die Schildkröte Strecken und Bögen laufen läßt und diese dabei eine Linie auf dem Schirm zeichnet. Ihre »Schildkröte« befindet sich immer zuerst in der Bildschirmmitte und kann dann mit einem einfachen aber sehr wirkungsvollen Befehlsvorrat gesteuert werden.
- **Grafik**
  Mit Hilfe verschiedener Befehle können Sie die Farbpalette umbenennen. Sie zeichnen einen Kreissektor oder einen Kreis und lesen beliebige Bildschirmausschnitte in einen Buffer ein. Diesen Ausschnitt können Sie nun an eine andere Bildschirmstelle placieren und mit Hilfe eines weiteren Befehls bestimmen Sie die Farbe irgendeines Punktes im Bildschirm. Mit der Prozedure »Pattern« ist eine ausgesprochen komplexe und freie Gestaltung möglich, einschließlich der Kolorierung von Bildschirmausschnitten.

** TM Micropro*

Neben vielen anderen neuen Features enthält Turbo-Pascal 3.0 für den Schneider-Rechner alle Funktionen, die ihn bereits in der 2.0 Version berühmt gemacht haben:

- *Vollwertiger Bildschirmeditor mit kompletter, menügesteuerter Entwicklungsunterstützung*
- *14-stellige Fließkomma-Arithmetik*
- *Diverse Zusatzfunktionen zur hardwarenahen Programmierung*
- *Dynamische Strings mit umfangreichen Handhabungsmöglichkeiten*
- *Program-Chaining mit gemeinsamen Variablen*
- *Sequentieller- oder Random-Zugriff auf Daten-Files*
- *Voller Zugriff auf alle Betriebssystem-Funktionen*
- *komplettes Overlay-System*
- *Deutsches Handbuch*

TURBO TUTOR™

Damit auch Pascal-Anfänger eine Chance haben mit Turbo-Pascal effektiv und professionell zu arbeiten, wurde von Borland ein Turbo-Tutor entwickelt, der sich als Lehrbuch versteht und trotzdem auch für den Experten in Sachen Pascal noch Tips und Tricks enthält. Sie erhalten ein Buch und eine Diskette, auf der alle Beispiele als Source-Code enthalten sind.

Zum Schluß noch die Stimme der Fachpresse:

CHIP
Die Kombination von Texteditor und Compiler macht die Fehlerbeseitigung, den wohl zeitaufwendigsten Teil der Programmierarbeit, technisch zum Kinderspiel.

Computer Persönlich
markiert 1984 eine neue Epoche der Benutzerfreundlichkeit: das Erscheinen eines spottbilligen Programmiersystems, das alles bisherige an flinker Leistungsfähigkeit und auf engstem Raum integrierter Benutzerführung in den Schatten stellt:
TURBO-PASCAL von BORLAND ...

c't
Turbo-Pascal dürfte auf dem besten Wege sein, zum Programm des Jahres gewählt zu werden ... und es juckt einem als Redakteur stets in den Fingern, den etablierten Software-Häusern unter die Nase zu reiben, daß gut und preiswert zwei Begriffe sind, die sich nicht gegenseitig ausschließen müssen.

Bitte hier schneiden!

Bestellcoupon:

☐ **Schneider CPC 464 Turbo-Pascal 3.0 Version I**	**DM 225,72**	(excl. MwSt.) **(DM 198,–)**
☐ **Schneider CPC 464 Turbo-Pascal 3.0 Version II mit Grafik-Erweiterung**	**DM 285,--**	(excl. MwSt.) **(DM 250,–)**
☐ **Turbo-Tutor**	**DM 104,86**	(excl. MwSt.) **(DM 98,–)**

☐ andere Rechner/Fabrikat: ____________

Kontrollieren Sie bitte, ob die Angaben korrekt sind.

Angaben zum Rechner:
☐ 8 Bit ☐ 16 Bit CPU ________ (z. B. »Z 80«)

Diskettengröße:
☐ 3 Zoll ☐ 3½ Zoll ☐ 5¼ Zoll ☐ 8 Zoll

Betriebssystem:
☐ CP/M 80 ☐ CP/M 86 ☐ MS-DOS ☐ PC-DOS

Fabrikat/Typ: ____________

Inland: ☐ Scheck (Versandkosten incl.)
☐ Nachnahme (+ DM 6,– Versandkosten)

Ausland: ☐ Scheck (+ DM 10,– Versandkosten)
☐ Nachnahme (+ DM 16,– Versandkosten)

Name: ____________

Straße: ____________

PLZ/Ort: ____________

Telefon: ____________

Unterschrift: ____________

HEIMSOETH software
Fraunhoferstr. 13/Pf. 140280
D-8000 München 5
Tel. 089/264060

Elektronik- & Computertage Saar

Vom 6. – 8. September findet in der Kongreßhalle Saarbrücken die 2. Elektronik- & Computertage als Verkaufs- und Informationsmesse statt.
Unter der Schirmherrschaft des saarländischen Landtagspräsident Albrecht Herold bietet sich ein repräsentativer Querschnitt durch die gesamte Elektronik- & Microcomputerbranche. Neben den kommerziellen Rechnern findet der Besucher eine große Auswahl an Zubehör und Fachliteratur. Aber auch Elektronik-, Audio- und Videotechnik gehören zum Messeschwerpunkt. Im Foyer West finden die jüngeren Besucher ihr "Heimcomputerparadies" mit allem, was das Herz begehrt. □

Der Computer-Arbeitsplatz

Speziell für die beiden Schneider Computer 464/664 entwickelte die Firma Zender + Reum einen Computer-Arbeitstisch. Neben einer ausziehbaren Rechnerauflage bietet dieses Möbelstück eine drehbare Monitorauflage, die mit maximal 80 kg belastet werden kann. Selbstverständlich finden auch Peripheriegeräte, wie Drucker und Floppy, ihren Platz auf dem Arbeitstisch. Gegen einen Aufpreis sind zusätzlich auch die Verlängerungskabel für Monitor und Rechner erhältlich. Die am Monitor ausgeführten Kabel reichen in Verbindung mit diesem Möbel nicht aus.
Der in schönem Design gehaltene Computer-Arbeitsplatz ist zum Preis von ca. DM 182,– erhältlich.
Info:
Zender + Reum
Richard-Wagner Straße 17
6600 Saarbrücken □

Microbuffer

Bei vielen der auf dem Markt angebotenen Drucker vermißte man bisher einen Zwischenspeicher, der die ankommenden Daten "buffert" und somit den Rechner für weiterführende Arbeiten freistellt. Als Beispiel mag hier der Schneider Drucker NLQ 401 dienen, der leider keinen eingebauten Buffer besitzt, so daß während eines Druckvorgangs der Computer blockiert ist. Diesem Manko kann auf zwei Wegen abgeholfen werden: Zunächst besteht die Möglichkeit, einen solchen Zwischenspeicher softwaremäßig zu simulieren. Das funktioniert zwar recht gut, geht jedoch voll auf Kosten des freien Speicherplatzes und ist somit für professionelle Anwendungen kaum geeignet. Mit dem Microbuffer der Fa. Inmac hat man nun eine hardwaremäßige Lösung dieses Problems gefunden. Der Microbuffer wird einfach zwischen Computer und Drucker geschaltet und sofort steht ein entsprechender Zwischenspeicher für die ausgehenden Daten zur Verfügung. Der Microbuffer wird in zwei verschiedenen Ausführungen geliefert, der Bufferspeicher beträgt entweder 32 KB oder 64 KB. Die externe Stromversorgung dieses Gerätes garantiert das fehlerfreie Arbeiten ohne eventuellen Systemabsturz. Der Microbuffer ist kompatibel zu allen Microcomputern und Druckern mit paralleler Schnittstelle, eine serielle Ausführung des Zwischenspeichers wird ebenfalls angeboten. Der Preis des Microbuffers beträgt ca. DM 1.070,–, was ihn wohl eher in den Profi-Bereich rücken läßt. Optional kann eine Speichererweiterung bis zu 256 KB erworben werden, die per RAM-Bausteinen nachgerüstet wird.
Info:
Inmac
Frankfurter Str. 103
6096 Raunheim □

MP-1 endlich lieferbar

Besitzer des CPC 464 mit Grünmonitor haben lange warten müssen. Nun endlich, nach mehrfacher Überarbeitung, ist das Netzteil MP-1 zum Anschluß des 464 an einen Farbfernseher lieferbar. Das Netzteil ist mit eigener Stromversorgung versehen und wird einfach zwischen Rechner und Fernseher gesteckt. Gerade für die farbliche Darstellung der zahlreichen Computerspiele ist das MP-1 interessant, diese gewinnen nun erheblich an Attraktivität.
Die Qualität des wiedergegebenen Fernsehbildes ist durchweg gut. Lediglich bei der 80-Zeichen-Darstellung von Text gibt es Abweichungen vom Monitorbild, was aufgrund der geringen Auflösung eines Fernsehbildschirmes nicht verwunderlich ist.

Info:
Schneider,
Türkheim □

C für CPC

Die Programmiersprache ‚C', bisher nur den teuren PC's vorbehalten, gibt es jetzt in einer Version für den Schneider Computer CPC 464.
‚C' unterstützt vor allem die Programmierung komfortabler Datenbanken und weist Pascal-ähnliche Strukturen auf. Übrigens wurde das weltweit bekannte Betriebssystem „UNIX" zu 90% mit ‚C' erstellt, ein Indiz für die enormen Möglichkeiten, die einem ‚C'-Programmierer zur Verfügung stehen. Das Programm gibt es momentan nur in England, doch wird es bei uns wohl nicht lange auf sich warten lassen.
Info:
Hisoft,
180 High Street North,
Dunstable,
Beds LU6 1AT □

Computer-Flohmarkt

Am 22. September 1985 findet in der Stadthalle Wuppertal der 2. Bergische Computer-Flohmarkt statt.
Auf ca. 900 qm Ausstellungsfläche wird alles angeboten, was mit Computern zu tun hat. Damit ist dies der z.Z. größte Computer-Flohmarkt Deutschlands, der seine Pforten am Sonntag den 22.9.85 von 11.00-17.00 Uhr geöffnet hat.
Der in diesem Jahr bereits zum zweitenmal durchgeführte Computer-Flohmarkt erfreut sich reger Nachfrage. Bei der im Januar abgehaltenen Veranstaltung waren weit über 1000 Besucher registriert worden – man erwartet jedoch eine deutliche Steigerung der Aussteller- und Besucherzahlen. Übrigens sind auch Händler bei dieser Ausstellung zugelassen.
Info:
Buchhandlung Finke
Postfach 132252
5600 Wuppertal 1 □



Haben Sie Fragen?

Trotz zahlreicher Literatur- und Programmauswahl gibt es Momente, in denen vielleicht einige Probleme auftreten, und Sie vor einem Rätsel stehen. Scheuen Sie sich nicht, uns mit Ihren Fragen zu konfrontieren. Wenn Sie zu irgendwelchen Themen, einem Programm oder einer Routine Fragen haben, teilen Sie uns diese getrost mit. Gerade bei denjenigen, die sich noch nicht lange mit Computern beschäftigen, tauchen oft Probleme und Fragen zu diesem Thema auf. Mit dieser Rubrik möchten wir Kontakte zu unseren Leser herstellen und ihnen die Gelegenheit geben, Fragen, Probleme, Tips und Kritiken loszuwerden.
Wir werden versuchen, Ihre Fragen – und seien sie auch noch so harmlos – fachgerecht zu beantworten.
Wir freuen uns auf Ihre Mitarbeit und hoffen, jedem ein guter Partner zu werden.

Ihre Redaktion

PS: Die Redaktion behält sich vor, aus Platzgründen Leserzuschriften in gekürzter Form wiederzugeben.

Textverarbeitung

Für meinen CPC 664 bin ich auf der Suche nach einem professionellen Textverarbeitungsprogramm. Zur Zeit habe ich zwei Programme in die engere Wahl gezogen und zwar Texpack von Schneider und Phase 4 von Brain-Computing. Allerdings konnte ich beim Ausprobieren dieser Programme im Fachgeschäft keine Unterschiede feststellen. Sind diese Programme identisch?

K. Häder, Augsburg

CPC:
Phase 4 und Texpack sind die gleichen Programme. Texpack hat lediglich eine Option implementiert, die Daten des Anwendungspaket „Compack" übernehmen kann.

Probleme mit Memory

Bei meinem Rechner scheint der Memory-Befehl nicht richtig zu funktionieren. Gebe ich zum Beispiel Memory &A00 ein und dann die Anweisung Load", so erscheint die Fehlermeldung Memory Full. Dabei erscheint weder der Schriftzug Press Play... noch wird Load überhaupt durchgeführt. Da dies auch bei anderen Schneider Rechnern passiert, stellt sich die Frage, ob hier wohl ein Hardware-Fehler vorliegt. Was ist also am Memory-Befehl nicht in Ordnung?

Bernd Müller-Dautenheimer, Ulm

CPC:
Der Memory-Befehl arbeitet einwandfrei. Dies wird allein daraus ersichtlich, daß er gerade bei Ladeanweisungen diese Fehlermeldung ausgibt. Setzen Sie die Memory nämlich unter 5000, so schneiden Sie den Laderoutinen den Speicherplatz ab, den diese zum Erzeugen Ihres Ladebuffers benötigen.
Wenn Sie etwas in den Bereich &A00 laden wollen, so geben Sie folgendes ein:
Memory 5000
Load "Name",&A00

Formatieren

Wie ich gehört habe, kann man Disketten auf drei verschiedene Arten formatieren (IBM-Format, Data-Format und CP/M-Format). Ich würde gern wissen, wie man das macht und ob das von Amsdos aus geht?

Boris Zech, Hannover

CPC:
Das Formatieren von Disketten kann nur unter CP/M vorgenommen werden. Zur Formatierung, im gewünschten Format, hängt man einfach das entsprechende Kürzel an den Format-Befehl an. Geben Sie die CP/M-Diskette in das Laufwerk und schreiben Sie wie folgt:

I CP/M
A> Format (für CP/-Format)
oder
A> Format D (für Data-Format)
oder
A> Format I (für IBM-Format)

Minispreadsheet

Hallo Minispreadsheet-Verschönerer,
wenn seit CPC International 7/85 das Kind (die Datei) einen Namen hat, dann schreibt ihn doch auch rein!

```
430 z$=" Einnahmen-/Ausgabenübersicht "+mon$
    (monl) +mon$ (mon2-(12 and mon2>12))+
    ":"+name$+" "
450 z$" Einnahmen-/Ausgabenübersicht "+mon
    $(monl)+":"+name$+" "
```

R. Günther, Düsseldorf

Erfahrungsbericht Hisoft-Pascal

Mein erstes Zusammentreffen mit dem Hisoft-Pascal-Compiler von Schneider war niederschmetternd:
Ein Ordner mit 50 Seiten Papier und eine beigelegte Cassette..., das soll 199,– DM wert sein?

Also wurde das Programm erst einmal geladen, was mich etwas versöhnte, denn die Ladezeit von 4:30 Minuten finde ich zufriedenstellend.
Der Editor-Modus war zuerst etwas ungewohnt, denn er kennt ja keine Direkt-Befehle, so daß Versuche, die Grundfarben blau und gelb zu ändern, erfolglos blieben.
Die Editor-Befehle sind jedoch einfach zu bedienen (1 Zeichen) und (abgesehen davon, daß sie englische Bezeichnungen haben) schnell und übersichtlich.
Nachdem mein erstes Programm nun stand, wurde compiliert und so lernte ich den (einzigen) Haken des Compilers kennen: Fehlermeldungen werden auch zur Laufzeit nur mit Nummern angegeben, so daß ein dauernder Blick in das Handbuch nötig war.
Nun jedoch wurde ich positiv überrascht, denn der Compiler war sehr schnell und auch die Ausführgeschwindigkeit des Programms war hervorragend. Hier nun wurde ich tätig und stellte einige Programmbeispiele auf, um die Schnelligkeit des Hisoft-Pascal zu messen.

Hier mein Ergebnis:

Art des Programms:	**Größe der Arrays:** 1-100	1-500	1-1000
Bubblesort (basic)	1:18,32	über 20 M.	-
Bubblesort (pascal)	3,18	1:19,37	-
Shell-Metzner-Sort (basic)	8,95	1:08,70	-
Shell-Metzner-Sort (pascal)	0,23	1,70	3,84
Quicksort (basic)	6,60	35,44	-
Quicksort (pascal)	0,36	1,87	4,02

Die Zahlen in den Arrays (Feldern) waren umgekehrt sortiert und die Berechnungen erfolgten in Sekunden.
Die Schnelligkeit ist in Pascal somit im Schnitt um das 25igfache größer. Noch zwei kurze Beispiele:
Es wurden
a) die Quadrate der ersten 5000 Zahlen
und
b) die Wurzeln der ersten 5000 Zahlen berechnet.

zu a)
Basic 2:34,98 sek.,
Pascal 17,23 sek.
(9-fache Geschwindigkeit)
zu b)
Basic 21,86 sek.,
Pascal 4,31 sek.
(5-fache Geschwindigkeit).

Kurze Zusammenfassung:
Der Compiler erfüllt meine Erwartungen im Hinblick auf Schnelligkeit des Compiliervorgangs und des Programmablaufs.
Das Handbuch ist zwar in den Befehlsbeschreibungen zu knapp gehalten, jedoch die im Anhang zu findenden Hinweise zum Aufruf der Firmware (somit ist eben auch Farbgestaltung und Grafikmodus möglich) sind sehr nützlich.
Zu erwähnen ist noch der Befehlsumfang des Hisoft-Pascal, der ähnlich wie das Schneider Basic über den Umfang eines Standards hinausgeht, denn auch im Hisoft-Pascal sind den Schneider Programmierern so liebgewonnene Hilfsmittel wie SOUND, EVERY, AFTER, POKE, PEEK und Grafikbefehle erhalten geblieben.
Das Hisoft-Pascal entspricht zudem dem Standard-Pascal, mit der Ausnahme, daß Record-Variablen nur vom einfachen Typ sein können.
Alles in allem sind meine Erwartungen, die ich als Informatik-Student gestellt habe, durch den CPC sowie durch das Hisoft-Pascal erfüllt worden!

Michael Klie, Wiehl-Bielstein

CPC:
Für diesen tollen Erfahrungsbericht bedanken wir uns herzlich. Wer seine Erfahrungen mit bestimmten Produkten loswerden möchte, ist hiermit zur Einsendung entsprechender Beiträge aufgerufen!

Druckeranpassung

Leserbrief Ausgabe 8, Seite 8, Herr Ditton.
Wir beziehen uns auf Ihren Leserbrief in der Ausgabe 8 der Fachzeitschrift CPC Schneider International und teilen Ihnen folgendes zu Ihrer Frage mit:

Generell können alle Epson-kompatiblen IMPACT DOT PRINTER Steuerzeichen durch Benutzen der CONTROL KEY's verstehen und ausführen.
Es ist aber zu beachten, daß nicht alle Steuercodes, die in der CONTROL KEY-TABELLE des entsprechenden Bedienungshandbuches zu Ihrem Drucker aufgeführt sind, unbedingt den gewünschten Erfolg bringen.
Einige Hersteller beschreiben in ihren Handbüchern, daß z.B. die Control-Codes dezimal (0,11,12,18,20) etc. Störungen bei der Ausführung erzeugen können.
Diese sind durch ein vorangestelltes %-Zeichen gekennzeichnet.
Das Problem weitet sich aber insbesonders bei einer Konfiguration CPC und Drucker aus, wenn bestimmte Codes innerhalb eines Strings auf die Platte geschrieben werden sollen.

Diese Zeichen sind auf dem Bildschirm dann nicht mehr darstellbar, können unter anderem zum Absturz führen oder verändern den String im günstigen, aber nicht gewünschten Falle.

Das übliche Kommando zum Unterstreichen eines Strings lautet bei Epson-kompatiblen Druckern z.B. wie folgt:

Print # 8,chr$(27);chr$(1)
zum Einschalten der Funktion,
Print # 8,chr$(27);chr$(0)
zum Ausschalten der Funktion.

Wenn die Codes 0 und 1 auf die Platte geschrieben werden sollen, wird nach dem Einladen der Datei die gewünschte Funktion bei Verwendung einer Stringkette nicht ausgeführt.
Ein ähnliches Problem tritt bei dem PRINT oder WRITE-Kommando auf, wenn z.B. Satzzeichen abgelegt werden sollen.

Eine Verwendung mittels Standard Control-Key's durch direkte Tastatureingabe ist also mit Vorsicht zu genießen.
Abhilfe läßt sich durch einen Code-Transfer über ASCII 128 und durch Stringaddition erreichen.
Die Control-Codes werden einer Variablen zugeordnet und zum eigentlichen Textstring addiert.

Beispiel:
x$=chr$(155)+chr$(173)+chr$(129)
y$=chr$(155)+chr$(172)+chr$(128)

Der Ausgabestring wird dann wie folgt generiert:

x$ Unterstreichen einschalten
y$ Unterstreichen ausschalten
Text$ Wort, Satz oder Buchstabe
z$ Ausgabestring codiert
z$ = x$ + Text$ + y$

Diese Methode verlangt auch nach einem neuen Konzept für die Druckausgabe schlechthin.
Es lassen sich aber somit sehr einfach codierte Texte auf eine Datei ohne Nebeneffekte ablegen, desgleichen ist eine Darstellung auf dem Bildschirm mit den Sonderzeichen möglich.
Vergleichen Sie bitte auch Literaturhinweise zum Beispiel zu den verbotenen Zeichen (Seite 265 – 266/ Floppy Buch).
Der CPC ist kein Apple, die Eigenarten des CPC machen manchmal Kompromisse notwendig.
Wenn auch der besondere Zeichensatz des CPC unter Umständen verwundert, so sind hierbei seine Sonderzeichen über Code 128 zumindest sehr hilfreich.

F. Werder, Hamburg

Joystick-Abfrage

Ich möchte wissen, wie ich eine einfache Joystick-Steuerung herstellen kann.

Lutz Kamrath, Hamburg

CPC:

```
10 a=JOY(0)
20 IF a=4 THEN XPOS=
XPOS-1:' bewegung na
ch links
30 IF a=8 THEN XPOS=
XPOS+1:' bewegung na
ch rechts
40 IF a=1 THEN YPOS=
YPOS+1:' bewegung na
ch oben
50 IF a=2 THEN YPOS=
YPOS-1:' bewegung na
ch unten
60 IF a=16 THEN fire
=1     :' feuerknopf
ist gedrueckt
```

Dauerfeuer

Vor kurzem legte ich mir den Quickshot II Joystick zu, der per Schalter auf Dauerfeuer eingestellt werden kann. Leider mußte ich feststellen, daß diese Einrichtung in Verbindung mit dem CPC 464 nicht funktioniert. Zuerst dachte ich, daß mein Computer defekt ist, doch auch auf anderen CPC Rechnern funktionierte das Dauerfeuer nicht, auch sonst kommen unverständliche Zeichen beim Drucken der Feuertaste auf den Bildschirm. Woran liegt das?

B. Städing, Karlsruhe

CPC:
Leider ist es nicht möglich, die Dauerfeuereinrichtung mit dem CPC zu nutzen. Ohne Einschränkung funktioniert der Quickshot II nach Durchtrennen der Dauerfeuereinrichtung. Dabei müssen Sie folgendermaßen vorgehen:

1. Die rote Litze, die aus dem Anschlußkabel kommt, von der Platinensteckfläche '+5V' abziehen und auf den leeren Platz daneben stekken (das rote Kabel kommt vom 9-poligen Stecker von Pin 6, Feuer 2).
2. Beim Dauerfeuerumschalter die gelbe Litze am mittleren Kontakt ablöten und mit der orangen Litze daneben verlöten.

Play and Poke

Hiermit möchte ich gern auf den Leserbrief aus Heft 7/85, mit der Überschrift „Pokes", der leider ohne Namensangabe abgedruckt wurde, eingehen. Die folgenden Pokes werden von mir benutzt:

1. Pyjamarama
10 Memory 8191
20 Load ""
30 Poke 16087,0
50 call 8192
2. Manic Miner
Poke & 6FA9,0
3. Punchy
10 Memory & 1FFF
20 Load "Code"
30 Poke & 20A9,255
40 call & 2000
4. Roland on the ropes
10 Memory 4800
20 Load "Roland.d"
30 Load "Roland.c"
40 Poke -25804,0:Poke -25562,0
50 call 41100
5. Defend or Die
10 Memory &3FFF
20 Load "Defend or Die"
30 Poke & 64e4,&99
40 Poke & 64e9,&99
50 call & 4025
6. Roland in time
Memory 4999
Load "Rointime", 5000
wenn geladen:
Poke 5650,167:call 5000
7. Electro Freddy
10 Memory 10000
20 Load "Al":Load""
30 Load "": Load"":Load
40 Poke 39356,255
50 call 39323
8. Android 1
Poke & 6391,&99
9. Jetset Willy
10 Mode 1:Memory & 7FED
30 Load"
40 For n = 170 TO 244
50 Read a$
60 Poke n, valC"&"+a$)
70 Next
80 CIS
90 call 170
100 Data 3e,32,80,81,21,00, 80,11,01,80,01,05,00,36,00, ed,b0,cd,ee,7f,01,00,15,21, 20,20,22,75,af,c5,21,5d,af
110 Data cd,53,af,78,58,87, 87,87,47,7b,c6,05,16,08,af, 05,f2,c8,00,3e,7f,32,f0,81, c3,4A,AC

Zum Schluß noch ein Problem, das mich beschäftigt: Im Adventure „Message from Andromeda" bekomme ich den Schlüssel nicht in die Computerkonsole. Wer kann mir helfen?

Ulf Lorenz, Hannover

CPC:
Mit diesen interessanten Pokes hat Herr Lorenz bestimmt vielen begeisterten Computerspielern sehr geholfen. Nun, wer hilft ihm bei seinem Problem?



CPC 6128: Der neue 128K Comput

Der erst kürzlich auf der CES in Chicago vorgestellte Amstrad CPC 6128 (CPC berichtete in Heft 7/85) wird nun doch vom Distributor Schneider vertrieben. Entgegen ursprünglichen Äußerungen wird der 128K Rechner schon sehr bald im Handel zu haben sein, denn der neue Schneider Computer wird bereits auf der Internationalen Funkausstellung in Berlin der breiten Öffentlichkeit präsentiert. Anschließend werden die ersten Geräte ausgeliefert, so daß ab Mitte September der CPC 6128 überall zu kaufen ist. In diesem Zusammenhang stellt sich natürlich die Frage nach dem Preis für diesen leistungsfähigen Rechner. Dieser ist eigentlich eine kleine Sensation und bleibt dem Konzept der Fa. Schneider treu - große Leistung mit niedrigen Preisen zu verbinden - kostet der CPC 6128 mit Grünmonitor doch nur ca. DM 1600,- und mit Farbmonitor DM 2100,-!! Wenn man nun die Preise des 664 mit denen des 6128 vergleicht, stellt sich sehr schnell der Preisunterschied von nur DM 100,- heraus. Für einen Aufpreis von DM 100,- erhält man also einen völlig neu gestylten CPC 664 mit doppelter Speicherkapazität. Damit setzt Schneider erneut einen Meilenstein, diesmal in der Kategorie der 128K Rechner und muß sich nun gegen Konkurrenten wie etwa den Commodore 128 behaupten. Aufgrund des enormen Preis-/Leistungsverhältnisses ist allerdings eine erfolgreiche Plazierung auf dem reichhaltig gesäten Computermarkt quasi vorprogrammiert, kann doch zur Zeit kein anderer Rechner mit solchen Leistungsdaten und Preisen aufwarten. Mit dem 664 und nun mit dem 6128 bringt Schneider in kürzester Zeit seit Markteinführung des CPC 464 im Herbst 1984 zwei neue 8-bit Computer mit exzellenten Leistungsdaten und nahezu konkurrenzlosen Preisen.

Was ist neu am 6128?

Der CPC 6128 präsentiert sich zunächst optisch in der gewohnten kompakten Bauweise mit integriertem 3"-Diskettenlaufwerk. Auf den ersten Blick fällt die veränderte Tastatur ins Auge, die in dieser Form und Tastenanordnung als professionell gelten muß. Die Anordnung der Tasten entspricht den Keyboards wesentlich teurerer Personalcomputer und unterscheidet sich deutlich von der des CPC 664. Beim Arbeiten am Gerät ist ein deutlicher Druckpunkt zu spüren, die Tasten sind jeweils nach außen gewölbt und sehr leicht zu bedienen.
Die Cursortasten wurden in den Funktionstastenblock eingebunden und sind dementsprechend nicht mehr so protzig wie wir das beim CPC 664 gewohnt sind. Die zum Editieren wichtige Copy-Taste befindet sich nun links neben der großen Space-Taste, gleich daneben sitzt die Control-Taste. Hier scheint sich allerdings ein Schwachpunkt aufzutun, denn die für viele Funktionen benötigte Shift-Taste ist nur umständlich zu erreichen. Allerdings benötigt man nur kurze Zeit, um sich an die geänderten Begebenheiten zu gewöhnen. Ein weiterer Pluspunkt ist die zweite Enter-Taste, die sich rechts neben der Space-Taste befindet und leicht zugänglich ist. Abschließend läßt sich sagen, daß diese professionelle Tastatur hohen Ansprüchen genügt und bei Rechnern dieser Preisklasse seines gleichen sucht.

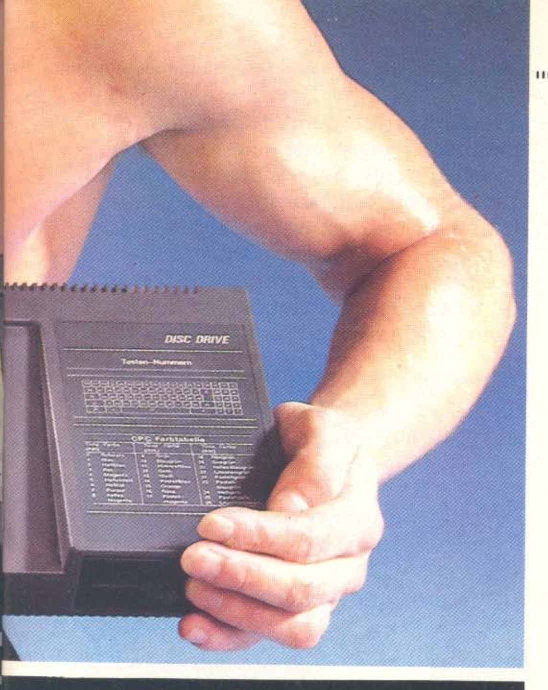
Bleiben wir zunächst bei den optischen Merkmalen des CPC 6128. Das Gehäuse ist wesentlich flacher geworden und hat entgegen dem 664 nur eine leichte Schrägstellung. Das integrierte Floppy-Laufwerk wurde den neuen Begebenheiten angepaßt und ist um ca. 5 cm flacher geworden. Diese kompakte und ergonomische Form läßt das Arbeiten mit dem 6128 zum Kinderspiel und Vergnügen werden. Die Code-Nummern für Tasten und Farben sind wie beim 664 auf der Oberseite des Disk-Laufwerkes aufgebracht, das erspart das Blättern im Bedienungshandbuch.

Wenden wir uns nun den rückwärtigen Anschlüssen und Bedienungselementen zu. Der erste Unterschied fällt schon beim Einschaltversuch des 6128 auf, denn sämtliche Bedienungselemente wie Einschalter, Lautstärkenregler sind von dem rechten Seitenteil verschwunden und befinden sich nun an der Rückseite des Computers.

Auch beim 6128 sind aufgrund des integrierten Disketten-Laufwerks drei Verbindungen zum Monitor notwendig, folglich können auch die Monitore des CPC 664 am neuen CPC betrieben werden.
Desweiteren befinden sich an der Rückseite die Anschlüsse für Drucker, Expansion Systems und zweites Disk-Laufwerk. Die Besonderheit hierbei ist, daß nicht wie bei den anderen Schneider Modellen die Platinen ausgeführt sind, sondern Centronics-Stecker zum Anschluß der Peripheriegeräte vorhanden sind. Diese Vorrichtung hat den großen Vorteil des funkentstörten Computerbetriebs, unliebsame Störungen bei TV- und Rundfunkempfang soll es also in Zukunft nicht mehr geben.
Die Anschlußports für Kassettenrekorder (DIN), Joystick und Stereoanlage befinden sich nunmehr an der linken Außenseite des Computers. Diese Anordnung erscheint uns sehr sinnvoll, überhaupt kann man das neue Konzept als überaus gelungen bezeichnen.

Technische Merkmale

Der CPC 6128 ist mit 128K RAM ausgestattet, die allerdings nur über das Bankswitching-Verfahren adressiert werden können (als Bankswitching bezeichnet man das Umschalten verschiedener Speicherbereiche). Da der Z-80 Mikroprozessor nur 64K zur gleichen Zeit verwalten kann, wird dieses Verfahren angewandt und der RAM-Speicher des 6128 in zwei Blöcken zu jeweils 64K unterteilt. Daher stehen nach dem Einschalten des Computers auch nur 42.249 Bytes (unter Basic) zur Verfügung, die restlichen Speicherstellen sind nur über Bankswitching zu erreichen.
Beim Arbeiten unter CP/M sieht die ganze Sache schon etwas anders aus. Dem CPC 6128 liegen die CP/M-Plus-Version sowie die LOGO-Plus-Version bei. Nach dem Booten von CP/M stehen dem Benutzer 61K zur freien Verfügung, entgegen den bei CPC 464 und 664 üblichen 39K. Damit sind die meisten CP/M-Programme wie Wordstar, dbase etc. uneingeschränkt lauffähig und der Anwendungsbereich des CPC 6128 erhöht sich um ein Vielfaches. Der Clou sind die auf den Systemdisketten befindlichen Utilities,



...er von Schneider

die eigentlich kein Programmierer missen möchte. Nachfolgend stellen wir Ihnen einige dieser nützlichen Programme vor:
Bankman – nach dem Laden dieses Files stehen für das Bankswitching einige RSX-Befehle zur Verfügung. Folgende Befehle sind möglich:

| Bankopen, Parameter
| Bankwrite, Parameter, Parameter
| Bankread, Parameter, Parameter
| Screenswap, Parameter, Parameter
| Bankfind, Parameter, Parameter
| Screencopy, Parameter, Parameter

SID - ruft den implementierten Monitor auf

Language – man kann zwischen 8 internationalen Zeichensätzen wählen

Disckit3 – Utility zum Formatieren und Kopieren einer Diskette

Date - Einstellen und Anzeigen des aktuellen Datums

Dump - zeigt die im Programm enthaltenen Variablen in Hex- und ASCII-Format

MAC - ruft den Macro-Assembler auf

XREF - zieht die Quersumme aller Variablen in einem Programm

CTRLCHARS - zeigt die Bedeutung der CTRL-Funktionen

SET24x80 – setzt den Bildschirm auf 24 x 80 Zeichen

Das waren nur einige der zahlreichen System-Utilities. In unserer nächsten Ausgabe werden wir auf diese noch im einzelnen eingehen.

Ein weiteres Dienstprogramm befindet sich auf der Rückseite der Logo-Diskette und trägt die Bezeichnung HELP. Dieses Programm erklärt einige der Utilities und gibt Anwendungsbeispiele sowie Syntax, leider wurde der Bankmanager für die Speicherbereichsverwaltung hier nicht berücksichtigt.

Software

Im Betriebssystem des CPC 6128 ist, wie beim 664, das Locomotive Basic der Version 1.1 eingebunden. Durch die Erhöhung der RAM-Kapazität und die daraus folgende Speicherverwaltung wurde das ROM des 6128 mit 48K ausgestattet (zum Vergleich: 464 und 664 haben je 32K ROM). Daraus ergibt sich die Verschiebung einiger Speicheradressen, so daß eine völlige Software-Kompatibilität mit dem 664 nicht gegeben ist. Allerdings gab es bei unseren zahlreichen Testversuchen nur ein einziges Programm, was nicht auf Anhieb lief und das ist der „deutsche Zeichensatz", Deutast von Schneider-Software. Ansonsten ist uns noch kein Programm bekannt, das nicht auf dem 6128 läuft.

Die Peripheriegeräte der CPC 464/664 sind vollständig kompatibel, der 7-bit Druckerausgang wurde also nicht geändert. Damit steht bereits zur Markteinführung eine große Palette an Peripherie und Software für den 6128 zur Verfügung, eine nicht unerhebliche Tatsache für ein erfolgreiches Abschneiden des neuen Schneider Personalcomputers.

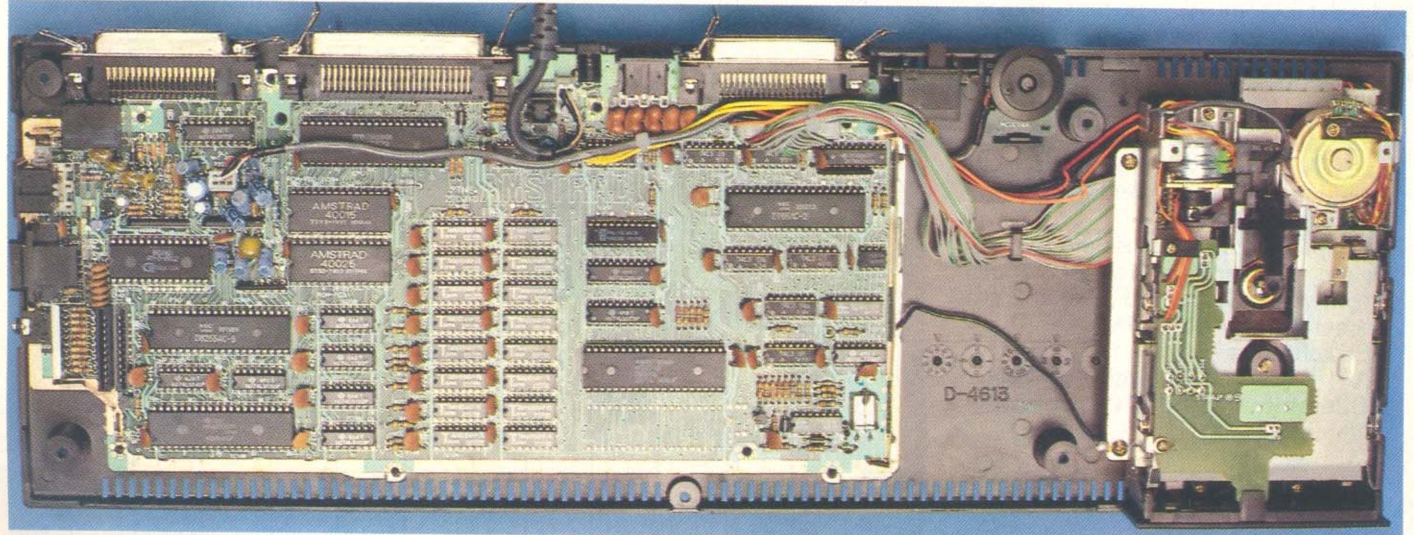
Fazit

Mit dem 6128 kommt ein Rechner auf den Markt, der allein schon wegen des Preis-/Leistungsverhältnisses ein Verkaufserfolg zu werden scheint. Aufgrund seiner enormen Leistungsdaten und der ausgereiften Konzeption drängt der CPC 6128 in die Kategorie der Personalcomputer und wird für professionelle Anwendungen interessant. Als Alternative zu wesentlich teureren Personalcomputern wie IBM, Apple etc. wird der 6128 seinen Weg auch in Klein- und Mittelunternehmen finden.

Daneben gibt es auch Unternehmen, die mit Computer-Anlagen ausgerüstet sind, deren Kapazität Sie bei weitem nicht ausnutzen können und hohe Wartungskosten tun ein übriges dazu. In solchen Fällen drängt sich der Einsatz des 6128 geradezu auf, denn zu einem Drittel des Preises vergleichbarer Rechner bietet er gleiche, wenn nicht sogar mehr Leistung.

Laut Auskunft der Fa. Schneider wird es demnächst eine 10 MB Festplatte für die CPC Rechner geben, deren Preis unter DM 3000,– liegen soll. Dann wird mit dem 6128 ein Profi-System vorliegen, das zum Preis einer Hobby-Anlage viele Freunde gewinnen wird.
Desweiteren bietet der 6128 für viele einen preiswerten Einstieg in die Welt der „großen PC's", man darf sich getrost auf diesen tollen Computer freuen.

(SR)

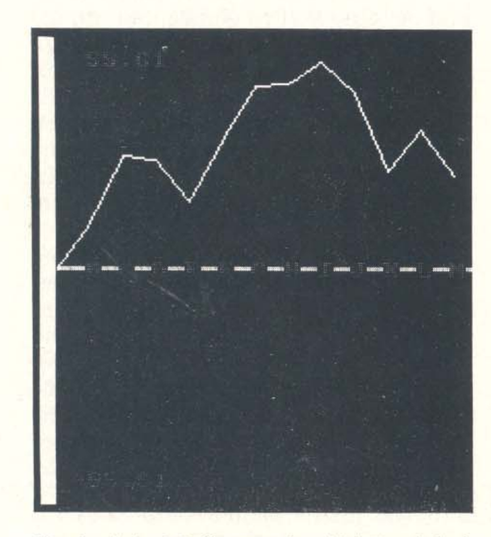


Auch das Innenleben des 6128 hat ein ansprechendes Design und präsentiert sich wesentlich kompakter, als vom 464 und 664 gewohnt. Deutlich zu erkennen sind die 16 RAM-Bausteine, die insgesamt eine Speicherkapazität von 128K RAM bieten. An der Rückseite sind die funkentstörten Centronics-Stecker sichtbar, mit denen ein absolut störungsfreier Computergenuß möglich sein soll.

Grafik-Kurs Teil 7

Diesmal heißt unser Hauptthema GRAFISCHE STATISTIK. Ein beliebtes Anwendungsgebiet, vor allem im kommerziellen Gebrauch. Es vereinfacht die Überschaubarkeit komplizierter Tabellen. Folgende Darstellungsmethoden sind allgemein üblich:

KURVENDIAGRAMM:



Sind viele Meßwerte in Abhängigkeit eines bestimmten Faktors bekannt, so wird oft die Darstellung in einem Kurvendiagramm benutzt. Beispiel:
Der Umsatz eines Lebensmittelgeschäfts soll in Abhängigkeit des jeweiligen Wochentages hervorgehoben werden.

Das größte Problem dabei ist die Bereichsüberschreitung. Doch dies wird nicht allzu schwer zu lösen sein, denn der CPC, mit seinem hervorragenden Basic, wird sich dabei als sehr nützlich erweisen.

Folgendes Vorgehen:

a) Errechnung des jeweiligen Abstandes (A) aus der Anzahl der zu besetzenden Punkte und der Gesamtbildschirmbreite.
Beispiel: Sechs Daten stehen zur Ausgabe. Es sind mit den Abständen vom Bildschirmrand sieben Zwischenräume erforderlich.
Errechnung:
A (= Zwischenraum) = 640 (= Bildschirmbreite)/(N (= Anzahl der Daten) + 1)/2

b) Nun muß das Maximum und das Minimum der Daten gesucht werden. Stehen die Variablen in dem Array DAT(), so läßt sich dieses Unterprogramm zur Suche verwenden, das Maxi- und Minimum zu finden:

c) Darstellung und Beschriftung einer Skala.

d) Errechnung der Bildschirmkoordinaten der einzelnen Meßwerte.

e) Die errechneten Punkte plotten und miteinander verbinden, eventuell

f) Vergleichsmöglichkeit einer anderen Farbe mit einblenden (Bsp.: Werte der Woche zuvor).

```
REM Max- und Minwert in DAT(N) suchen
GROSS = 0 FOR I=1 to N
GROSS = MAX(DAT(I),GROSS)
NEXT I
KLEIN = GROSS
FOR I=1 TO N
KLEIN = MIN(DAT(I),KLEIN)
NEXT I
RETURN
```

Und hier noch ein entsprechendes Basic-Programm, das durch kleine Änderungen in größeren Programmen als Unterroutine sehr nützlich eingesetzt werden kann:

Programm 1

```
100 REM Initialisierung
110 REM
120 CLS
130 INK 0,0 : PAPER 0
140 INK 1,25 : PEN 1
150 MODE 1
160 REM
170 N=7 : DIM DAT(N)
180 DATA 3,8,6,5,9,4,3
190 FOR I=1 TO N
200 READ DAT(I)
210 NEXT I
220 CLS
230 GROSS = 0
240 FOR I=1 TO N
250 GROSS = MAX(DAT(I),GROSS)
260 NEXT I
270 CLS
280 REM
290 REM Hauptschleife
300 REM
310 MASS = (400/(GROSS+1))/2
320 ANZ = (640/(N+1))/2
330 PLOT ANZ,MASS
340 DRAW 640-ANZ,MASS
350 PLOT ANZ,MASS
360 DRAW ANZ,400-MASS
370 PLOT ANZ,DAT(1)*MASS
380 FOR I=2 TO N
390 DRAW (I*2+1) * ANZ,MASS * DAT(I)
400 NEXT I
410 END
```

BALKENDIAGRAMME

Balkendiagramme beruhen auf dem gleichen Prinzip mit dem kleinen Unterschied, daß diese Werte noch etwas mehr zur Geltung kommen. Balkendiagramme werden meist bei einer kleineren Anzahl von Werten benutzt, als dies bei den Kurvendiagrammen der Fall ist. Unser Beispiel eignet sich dennoch recht gut, da sieben Werte nicht sehr viel sind.

Bis zu Punkt e) bleiben die Vorgänge gleich. Folgende Änderung sollte dann vorgenommen werden:

e) Von der O-Achse bis zur errechneten Höhe, Balken zeichnen.

Auch die eine entsprechende Unterroutine, die DAT(N) in Balken ausgibt.

Programm 2

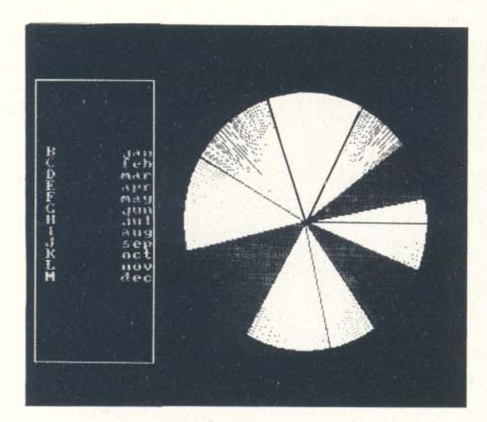
```
100 REM Initialisierung
110 REM
120 CLS
130 INK 0,0 : PAPER 0
140 INK 1,25 : PEN 1
150 MODE 1
160 REM
170 N=7 : DIM DAT(N)
180 DATA 3,8,6,5,9,4,3
190 FOR I=1 TO N
200 READ DAT(I)
210 NEXT I
220 CLS
230 GROSS = 0
240 FOR I=1 TO N
250 GROSS = MAX(DAT(I),GROSS)
260 NEXT I
270 CLS
280 REM
290 REM Hauptschleife
300 REM
310 MASS = 400/(GROSS+1)
320 ANZ = 640/(N+1)
330 FOR I=1 TO N
340 FOR J=1 TO ANZ/2
350 PLOT ANZ*I+J,MASS*DAT(I)+MASS/2
350 DRAW ANZ*I+J,MASS
370 NEXT J
380 NEXT I
390 END
```

KUCHENDIAGRAMME

Diese Art der grafischen Statistik ist geeignet, um Verteilungen besser darstellen zu können. Beispielsweise bei der Sitzverteilung im Bundestag usw...



Der ganze Kuchen (oder Kreis) entspricht dabei allen zu vergebenden Stimmen (100 %). Dann erhalten die Parteien nach Wahlausgang von den zu vergebenden Stimmen einen gewissen Anteil. Solch ein Anteil stellt dann ein Stück des Kuchens dar. Je größer ein Anteil ist, desto größer ist auch das Stück des Kuchens.



Doch wie ist dies zu programmieren? Dazu bedienen wir uns einfacher Mathematik.

Gegeben sei zum Beispiel:
100 Sitze sind zu vergeben. Davon erhalten:

Partei 1 - 40 Sitze
Partei 2 - 22 Sitze
Partei 3 - 21 Sitze
Partei 4 - 17 Sitze.

Mit einer einfachen Dreisatzrechnung läßt sich nun errechnen, wieviel des Kreises (wieviel von 360 Grad) die einzelne Partei erhält – nämlich:

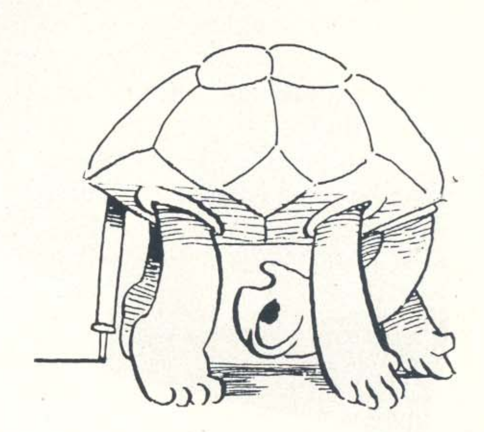
Partei 1 - 40 % entsprechend 144 Grad
Partei 2 - 22 % entsprechend 79 Grad
Partei 3 - 21 % entsprechend 76 Grad
Partei 4 - 17 % entsprechend 61 Grad

Nach einer Rundung muß darauf geachtet werden, daß nicht mehr Sitze vergeben werden, als vorhanden sind.

Und hier das entsprechende Listing:

Programm 3.

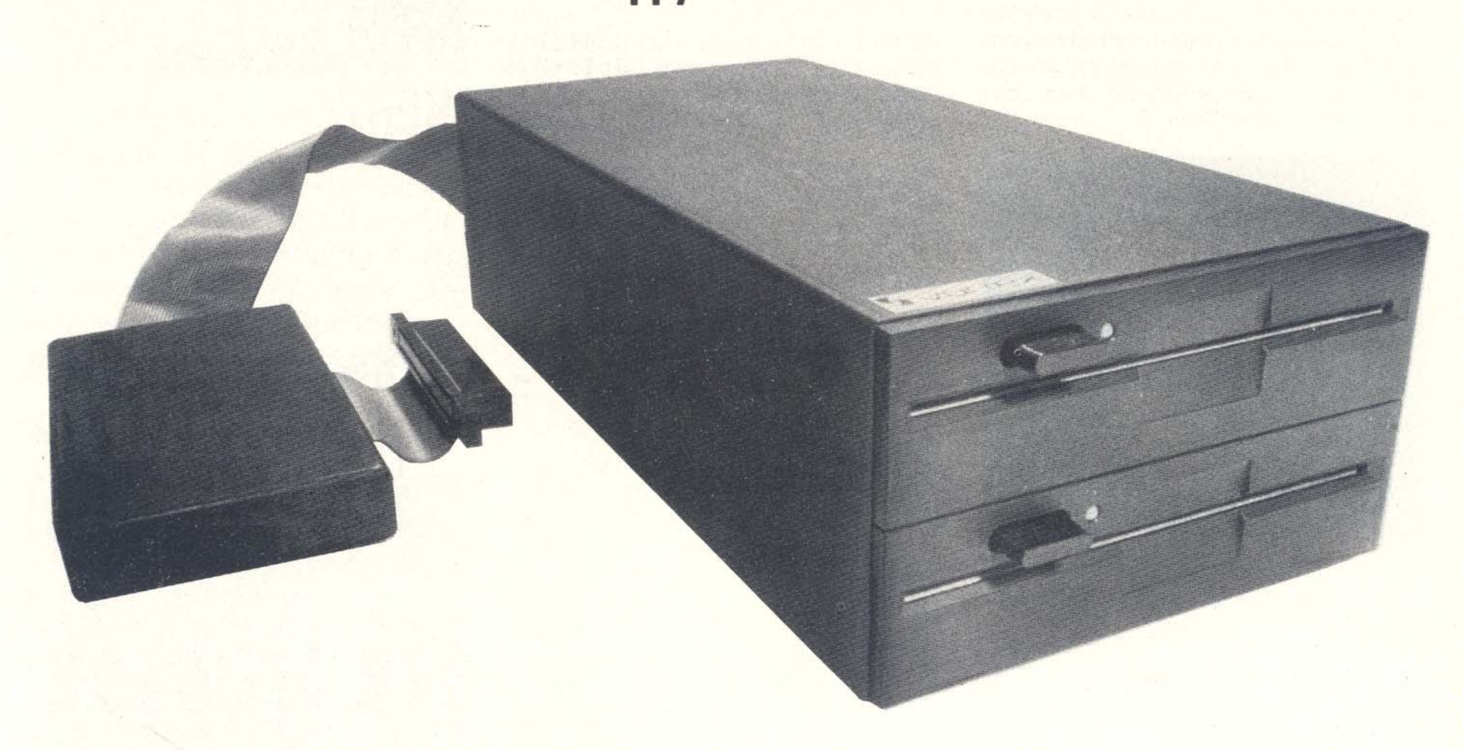
```
100 REM Initialisierung
110 REM
120 CLS
130 INK 0,0 : PAPER 0
140 INK 1,25 : PEN 1
150 MODE 1
160 REM
170 N=7 : DIM DAT(N)
180 DATA 3,8,6,5,9,4,3
190 FOR I=1 TO N
200 READ DAT(I)
210 NEXT I
220 CLS
230 GROSS = 0
240 FOR I=1 TO N
250 GROSS = GROSS + DAT(I)
260 NEXT I
270 CLS
280 DEG
290 V1=320
300 V2=200
310 A=180
320 REM
330 REM Hauptschleife
340 REM
350 FOR I=0 TO 359
360 PLOT A*COS(I) + V1,A*SIN(I) + V2
370 NEXT I
380 FOR I=1 TO N
390 DAT(I)=360*(GROSS/DAT(I))+DAT(I-1)
400 NEXT I
410 FOR I=1 TO N
420 X=A*COS(DAT(I)) + V1
430 Y=A*SIN(DAT(I)) + V2
440 PLOT V1,V2
450 DRAW X,Y
460 NEXT I
470 END
```



Im nächsten Teil unseres Grafik-Kurses werden wir dazu übergehen, LOGO anzuwenden. Viele von Ihnen, hauptsächlich die, die keine Floppy und dementsprechend auch kein LOGO besitzen, werden dies bedauern. Aber verstehen Sie bitte, daß auch diejenigen etwas Hilfestellung brauchen, die neu in einer Sprache beginnen. Bis zum nächsten Mal mit LOGO. (HF)

Das Laufwerk...

vortex Floppy Disk Station F 1



für den Schneider CPC 464

5.25" ..... 1,4MB .... CP/M 2.2

Leistungen

- ein (wahlweise zwei; von Anfang an, oder nachrüstbar) 5.25" Slimline 80 Track, DS/DD **6138 BASF** Laufwerk der modernsten Technologie mit 708KB (1,4MB) formatierter Speicherkapazität. 4 msec Steprate. IBM 34 Formate.
- CP/M 2.2 Betriebssystem und Systemutilities
- erweitertes BASIC – stand alone – Diskettenbetriebssystem VDOS
- ohne Soft- oder Hardwareänderungen kann ein Schneider 3"-Laufwerk über ein Adapterkabel angeschlossen werden. Softwarekonvertierung von 5.25" auf 3" und umgekehrt: Kein Problem.

Preise

- F 1/S Floppy Disk Station mit Controller und Laufwerk incl. CP/M 2.2, VDOS und Handbuch
  1198,– DM (unverbindliche Preisempfehlung)
- F 1/D Floppy Disk Station mit Controller und zwei Laufwerken incl. CP/M 2.2, VDOS und Handbuch
  1698,– DM (unverbindliche Preisempfehlung)

A1-S Aufrüstkit bestehend aus BASF Laufwerk 6138 und Einbauanleitung
500,– DM(unverbindliche Preisempfehlung)

5.25" Zweitlaufwerk F 1-Z

Das Laufwerk F 1-Z kann als Zweitlaufwerk an die Schneider 3"-Diskettenstation DDI-1 angeschlossen werden und hat dieselbe Speicherkapazität wie das 3"-Laufwerk. Es ist identisch mit Station F 1-S jedoch ohne Controller und ohne CP/M.
Das mitgelieferte Programm SPARA erlaubt Ihnen das Lesen und Beschreiben von Disketten gängiger CP/M-Systeme, welche Ihre Disketten einseitig mit 40 Spuren verwalten (diese Einschränkung bedingt der Controller der Schneider DDI-1).
Erwacht jedoch Ihr Interesse an 708KB oder 1.4MB, so können Sie Ihre F 1-Z problemlos zur F 1-S oder F 1-D aufrüsten.

Preise

- 5.25" Zweitlaufwerk F 1-Z + Programm SPARA
  698,– DM (unverbindliche Preisempfehlung)
- Aufrüstkit A 1-Z bestehend aus Controller, CP/M-Lizenz und Dienstprogramme sowie Handbuch
  548,– DM (unverbindliche Preisempfehlung)



Speichern Sie wohl...

...mit 64K bis 512K

zusätzlichem Arbeitsspeicher für Ihren CPC 464

vortex Speicherkarten...
... ein Muß, wenn Sie erleben wollen, was Ihr CPC wirklich kann:

- jetzt **endlich** läuft jedes **CP/M-Programm!** (z.B.: Wordstar, Multiplan, DBase...)
- **Druckerbuffer**, d.h., Sie können an Ihrem CPC weiter arbeiten, während der Drucker läuft.
- der Speicher ist auch unter Basic vollwertig als Daten-, Programm- und Bildspeicher einsetzbar.
- die Karte beinhaltet eine 128K ROM Erweiterung, welche die Steuerung des Speichers nahtlos in das bestehende Rechnerbetriebssystem einbindet. Dieser ROM beinhaltet außerdem mächtige Softwareergänzungen:
  - einen Systemmonitor (Debugger, Disassembler, Lineassembler)
  - Basicerweiterungen
  - für 3"-Floppy-Besitzer kann der Speicherkarten ROM mit VDOS 2.0 Features (z.B.: Relative Dateien geliefert werden).
- mit dem Aufstecken der Speicherkarte auf die CPC Grundplatine wird gleichzeitig eine System-Bus-Bufferung realisiert. Ihr CPC ist dadurch **endlich** für weitere **wirkungsvolle Peripherie** vorbereitet (z.B.: vortex 16Bit Co-Prozessor MSDOS Karte, RS 232 Karte, Expansion Karte usw.)
- ab der 128K Version können Sie den Speicher als sogenannte RAM-Disk betreiben und als Laufwerk 'C' ansprechen.

Jeder unserer Produkte liegt der vortex Service-Paß bei. Durch diesen garantieren wir Ihnen einen kostenfreien Anspruch auf alle Weiterentwicklungen und eventuelle Verbesserungen unserer Betriebssystemsoftware.

SP 64/M (keine ROM-Erweiterung, kein Bus-Buffer, nicht aufrüstbar)
DM 138,– (unverbindliche Preisempfehlung)

SP 64 (ROM-Erweiterung, Bus-Buffer, voll auf 512K aufrüstbar)
DM 275,- (unverbindliche Preisempfehlung)

Ausführliches Informationsmaterial bitte bei uns anfordern, wir informieren Sie gerne detaillierter. Beachten Sie auch hierzu unsere 'User-Sprechstunde'.

Sie erhalten unsere Produkte:

- in allen Karstadt und Horten Computercentern
- in den technischen Kaufhäusern Phora und Brinkmann
- im übrigen Fachhandel
- erfragen Sie weitere Bezugsmöglichkeiten bei uns.

Für soft- und hardware-technische Fragen, im Zusammenhang mit unseren Produkten, haben wir eine **'User-Sprechstunde'** eingerichtet. Montag und Donnerstag von 18 – 21 Uhr stehen wir Ihnen telefonisch zur Verfügung.



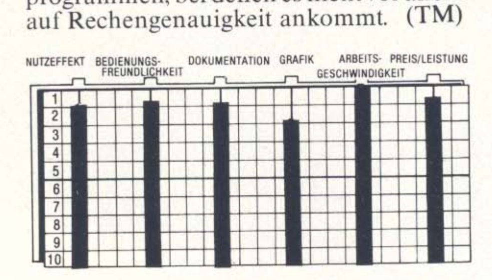
7106 Neuenstadt, Klingenberg 12, Abt.: Marketing Tel.: [illegible]



Speedbas CPC

Hersteller: Data Media
Vertrieb: Data Media
Steuerung: Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Unverb. Preisempfehlung: DM 89,– Cass./ DM 99,– Disk
CPC 464 ☒ CPC 664 ☐ CPC 6128 ☐

Basic-Programme haben den Vorteil, daß sie sehr leicht programmiert werden können, daß man sich über den Grundaufbau des Programmes nur grob umrissen im Klaren sein muß und daß nachträgliche Veränderungen sehr leicht durch Einfügen von Zeilen vorgenommen werden können.
Allerdings ist die Abarbeitungsgeschwindigkeit sehr niedrig. Gerade beim Spielen kommt dieses Manko besonders zum Vorschein.
Alternativ zu Basic kommen Compilersprachen wie Forth in Frage. Allerdings erfordern diese Sprachen wiederum genaue Programmierkenntnisse.
Einzige Möglichkeit für Programmierer, die den Weg über diese einfache Grundsprache gehen wollen, ist die Compilation des Basic-Programmes.
Ein solches Compilat hat viele Vorteile gegenüber einem Basic-Programm.
Zum einen wird das Programm sehr schnell, zum anderen ist es gleich gegen fremde Eingriffe geschützt.
Zwar entspricht die Geschwindigkeit nicht der eines reinen Assemblerprogramms, aber eine Geschwindigkeitssteigerung bis zu 200 mal schneller, die ist gegeben.
Der Data Media Compiler kann nahezu das ganze Locomotive-Basic compilieren. Einschränkungen sind durch die Tatsache gesetzt, daß das Programm nur mit Ganzzahlen arbeiten kann und Funktionen wie DEG und RAD, die ja sehr viel Kommastellen mitberechnen, nicht funktionieren können.
Auf Spiele haben diese Einschränkungen keine besondere Auswirkung. Die Integer-Rechnung äußert sich eher noch positiv, da Ganzzahlen wesentlich schneller bearbeitet werden als Floating Points.
Die Compilation erfolgt über Tastendruck und benötigt drei Durchgänge (Passes). Im ersten Pass wird die syntaktische Richtigkeit des Quellprogramm geprüft. Im zweiten die semantische Richtigkeit und die Vorausberechnung der Labels. Im dritten Pass findet die Umsetzung und Adressierung des Speichers statt. Wurde kein Fehler gefunden, fragt ein Menue ab, ob gestartet oder gesichert werden soll.
Da an das Programm eine 2KB große Runtime angehangen wird, ist Speicherung nur auf Kassette möglich. Hierbei wird jedoch eine spezielle Fast-Load-Routine implementiert, die die Kassettenoperation beschleunigt.
Über ein Kopierprogramm können die gesicherten Programme wieder in Standard-Kassettenformat umkopiert und dann auf Disk abgelegt werden.
Die Compilation erfolgt sehr schnell und beim Test des Programms konnte eine sehr große Geschwindigkeitssteigerung erreicht werden. Ein nützliches Utility für Programmierer von Spiel- und Anwenderprogrammen, bei denen es nicht vor allem auf Rechengenauigkeit ankommt. (TM)



Forth

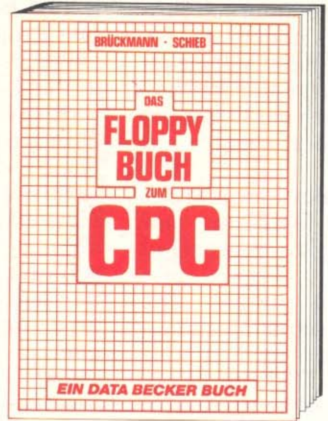
Hersteller: Forth-Systeme
Vertrieb: Forth-Systeme
Steuerung: Tastatur
Monitor: Farbe/Grün
Programm: 100% M-Code
Unverb. Preisempfehlung:
148,– DM Cass./168,– DM Disk
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Forth ist, wie Basic, Pascal oder Logo, eine Computersprache.
Sie unterscheidet sich jedoch sehr von den herkömmlichen Computersprachen.
Forth wurde im Jahre 1972 von Charles H. Moore entwickelt und zwar zur Steuerung des Teleskops am Kitt Peak Observatorium.
Schon damals waren die Hauptaufgaben von Forth die Lösung von Echtzeit-Problemen, komplexe Maschinensteuerungen sowie Steuerungsaufgaben im allgemeinen. Die Geschwindigkeit des Programmablaufs reicht nicht an die von Assemblersprache, ist jedoch höher als jede andere Programmiersprache.
Im Gegensatz zu Pascal ist Forth ein selbständiges Sprachsystem, in dem sowohl Interpreter, Compiler als auch ein eigenes Betriebssystem enthalten sind.
Einzigartig an Forth ist jedoch der Rechenweg, den jedes Programm nimmt. Nach dem Prinzip des Last In First Out Speichers, arbeitet Forth voll stapelorientiert. Auf diesen wird oben etwas aufgelegt, das später wieder herausgenommen wird.
Es gibt einen Stapel für Zahlen, auf dem die Berechnungen ausgeführt werden und den Stapel der definierten Wörter: das Wörterbuch.
Einen Zahlenstapel zu Berechnungen ist nun nicht gerade eine Sensation, jeder Hewlett Packard Taschenrechner arbeitet nach dem Stapelprinzip. Das völlig neuartige an Forth ist der Stapel für die Texte.
Die Sprache an sich besteht aus Worten, deren Definition, daß heißt, deren Festlegung, was sie bewirken sollen, auf dem Definitionsstapel (Wörterbuch) abgelegt werden. Der Start eines Programms erfolgt also durch die Eingabe eines einzigen Wortes.
Dieses Konzept führte dazu, daß Forth auch in sogenannten Translatoren (Sprachübersetzern) Verwendung findet.
Translatoren sind kleine, Taschenrechner-große Maschinen, in denen ein Wort eingegeben und die fremdsprachige Übersetzung ausgegeben wird.
Die Möglichkeit, sich die Worte selbst zu definieren, führt dazu, daß sich die Sprache laufend erweitert. Ein Forth Programm besteht also aus Worten, die in der Grundbibliothek des gekauften Programmes enthalten sind und weiteren, die man selber erstellt hat. Mit dem Befehl Forget können die Stapel jedoch auch wieder gelöscht werden.
Bei Rechenoperationen, die, wie schon erwähnt, auch über einen Stapel ablaufen, wird eine optimierende Mathematik angewandt, die REVERSE POLNISCHE NOTATION.
Dieses Rechenprinzip dürfte wieder allen Anwendern von HP-Taschenrechnern bekannt sein. Es hat den Vorteil, daß direkt mit dem Stapel über eine Postfix gerechnet wird. Dies bringt eine höhere Rechenleistung. Gibt man bei einer normalen Infix Rechnung z.B. 13+7 ein, so würde die gleiche Rechenoperation unter Postfix Notation : 13 7 + heißen.
Forthprogramme haben jedoch den Nachteil, daß sie nicht leicht zu beschreiben sind. Da die Wahl der Worte unbeschränkt ist, artet das Programmieren sehr leicht in wüstes Wirrwarr von Abkürzungen aus, und selbst der Programmierer findet sich nach einigen Tagen nicht mehr in seinem Listing zurecht, da er gar nicht mehr weiß, welches Wort nun welche Funktion hat. Dieser Nachteil von Forth wird jedoch durch eine ganze Reihe anderer Leistungsdaten wieder ausgeglichen. Was kann jetzt jedoch der Schneider Anwender damit anfangen, daß eine solch tolle Programmiersprache existiert, wie soll er das für seinen Rechner nutzen?
Nun, die Antwort ist einfach, denn seit kurzer Zeit bietet die deutsche Firma Forth-Systeme einen Forth Interpreter für den Schneider an.
Das Programm CPCFORTH kommt mit einem ca. 150 seitigem Handbuch und erlaubt Ihnen, den ganzen Leistungsumfang der Sprache zu nutzen.
Das System ist auf Disk und Kassette erhältlich und kann auf Kassette bis zu 30 gut gefüllte Screens gleichzeitig editieren und compilieren.
Durch die Verwendung von Trace-Funktionen und einer komfortablen Steuerung, lassen sich Programmabläufe leicht kontrollieren und optimieren. Interaktive Kommandos sind über die Eingabe von M möglich. So lassen sich beispielsweise Manipulationen am Stapel vornehmen, ohne daß Sie aus dem Trace Modus gehen müssen.
Definierte Worte können mittels des DECompilers wieder aufgeschlüsselt werden. Dabei brauchen Sie sich nicht zu merken, auf welcher Screen Sie nun die eigentliche Definition des Wortes stehen haben. Sie geben einfach "DIS NAME" ein und erhalten ein formatiertes Listing dieser Wortdefinition.
Eine breite Palette von Editorbefehlen erlauben die seitenweise Korrektur von Quelltexten und bietet Eigenschaften, wie beispielsweise ein Zeilenswap.
Tracer und DECompiler sind Hilfsmittel,

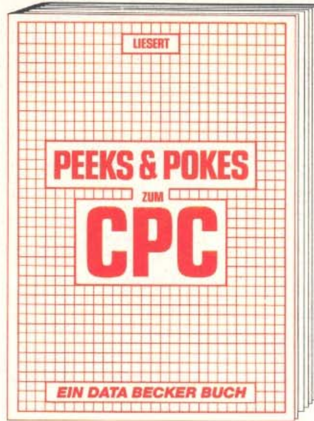
Neue Hits zum CPC



Endlich CP/M beherrschen – mit diesem Trainingsbuch kein Problem! Von ganz grundsätzlichen Erklärungen zu beispielsweise Speicherung von Zahlen, Schreibschutz oder ASCII-System über Schnittstellen, andere Betriebssysteme und Anpassung von CP/M bis hin zur ausführlichen Behandlung des Schwerpunktes Dateien. Außerdem CP/M-Hilfsprogramme und ihre Anwendung. Speziell zugeschnitten auf Ihren CPC.
CP/M Trainingsbuch zum CPC
ca. 250 Seiten, DM 49,–



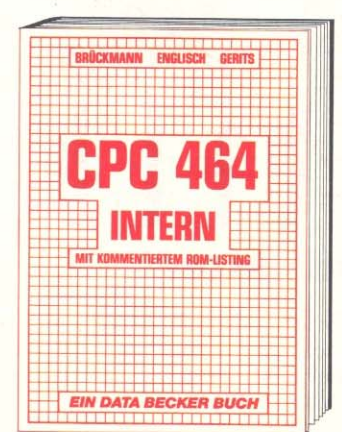
Alles über Diskettenprogrammierung vom Einsteiger bis zum Profi. Natürlich mit ausführlichem ROM-Listing (Betriebssystem), einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Monitor und einem ausgesprochen nützlichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen.
Das Floppy-Buch zum CPC
353 Seiten, DM 49,–



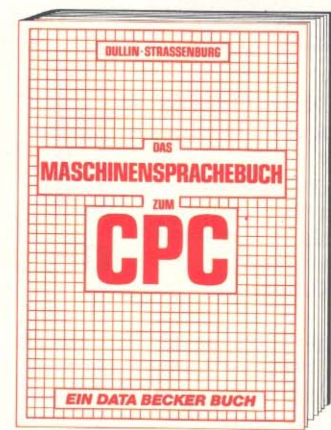
Wer die wichtigen Peeks und Pokes zum CPC kennen und anwenden will, der findet hier umfassende Information. Sie reicht vom Adreßbereich des Prozessors über Betriebssystem und Interpreter bis hin zur Einführung in die Maschinensprache. Dazu präzise Programmierhilfen, sinnvole Routinen sowie reichlich Material zu den Themen Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC, RAM-Pages.
Peeks und Pokes zum CPC
180 Seiten, DM 29,–



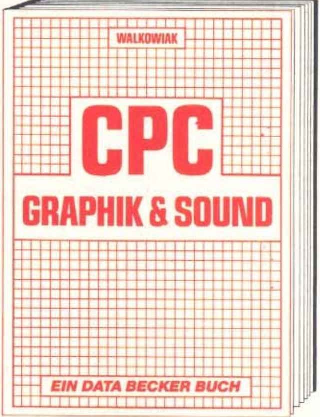
Speziell für den Hobbyelektroniker, der mehr aus seinem CPC machen möchte. Von nützlichen Tips zur Platinenherstellung über Adressdecodierung, Adapterkarten und Interfaces bis zu EPROM-Programmierboard und -Programmiernetzteil oder Motorsteuerung für Gleich- und Schrittschaltmotoren werden machbare Erweiterungen ausführlich und praxisnah beschrieben. Am besten gleich anfangen!
CPC Hardware-Erweiterungen
445 Seiten, DM 49,–



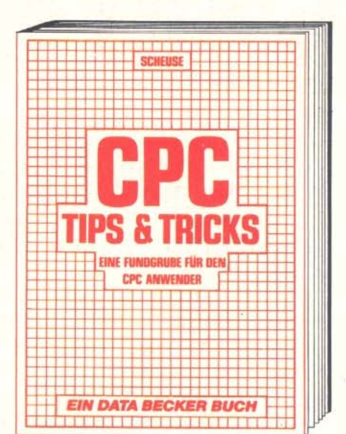
Wirklich alle Geheimnisse des CPC lüftet dieses Standardwerk, das für den Fortgeschrittenen BASIC-Programmierer unentbehrlich, für den Assembler-Programmierer ein absolutes Muß ist. Neben dem ausführlich dokumentierten und kommentierten BASIC-ROM-Listing enthält es umfangreiche Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z 80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinennutzung, Character-Generator, BASIC-Interpreter und mehr.
CPC 464 Intern mit kommentiertem ROM-Listing, 548 Seiten, DM 69,–



Von den Grundlagen der Maschinenspracheprogrammierung über die Arbeitsweise des Z 80-Prozessors und einer genauen Beschreibung seiner Befehle bis zur Benutzung von Systemroutinen ist alles ausführlich und mit vielen Beispielen erklärt. Im Buch enthalten sind Assembler, Disassembler und Monitor als komplette Anwenderprogramme. So wird der Einstieg in die Maschinensprache leichtgemacht!
Das Maschinensprachebuch zum CPC
330 Seiten, DM 39,–



In diesem erstklassigen Buch wird gezeigt, wie man die außergewöhnlichen Grafik- und Soundmöglichkeiten des CPC nutzt. Natürlich mit vielen interessanten Beispielen und nützlichen Hilfsprogrammen. Aus dem Inhalt: Grundlagen der Grafikprogrammierung, Sprites, Shapes und Strings, mehrfarbige Darstellungen, Koordinatentransformation, Verschiebungen, Drehungen, Rotation, 3-D-Funktionsplotter, CAD, Synthesizer, Miniorgel, Hüllkurven und vieles mehr.
CPC Graphik & Sound
220 Seiten, DM 39,–



Rund um den CPC viele Anregungen und wichtige Hilfen. Von Hardwareaufbau, Betriebssystem, BASIC-Tokens, Zeichnen mit dem Joystick, Anwendungen der Windowtechnologie und sehr vielen interessanten Programmen wie einer umfangreichen Dateiverwaltung, Soundeditor, komfortablem Zeichengenerator bis zu kompletten Listings spannender Spiele bietet das Buch eine Fülle von Möglichkeiten.
CPC Tips & Tricks
Eine Fundgrube für den CPC-Anwender
263 Seiten, DM 39,–

TEXTOMAT

Deutschlands meistverkaufte Textverarbeitung jetzt in einer speziellen Version für den CPC464 und 664. Erweitert um 80-Zeichen Darstellung, Tabulatoren, Word Wrap und Trennvorschläge. Natürlich mit deutschem Zeichensatz. Komplett in Maschinensprache und damit superschnell. Durch Menuesteuerung leicht zu bedienen. Läßt sich ideal mit DATAMAT kombinieren. **TEXTOMAT für den CPC kostet einschließlich umfangreichem Handbuch DM 148,–.**

DATAMAT

Deutschlands meistverkaufte Dateiverwaltung jetzt in einer speziellen Version für den CPC 464 und 664. Erweitert um 80-Zeichen Darstellung und größere Datensätze mit bis zu 512 Zeichen. Komplett in Maschinensprache und damit superschnell. Läßt sich ideal mit TEXTOMAT kombinieren. **DATAMAT für den CPC kostet einschließlich umfangreichem Handbuch DM 148,–.**

BUDGET-MANAGER

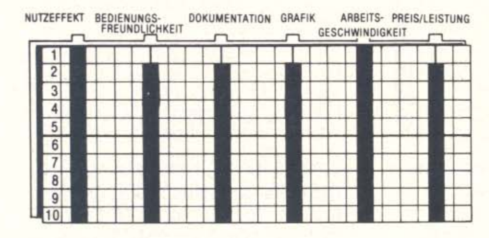
Universelle Buchführung sowohl für private Zwecke als auch zur Planung, Überwachung und Abwicklung von Budgets jeglicher Art. **BUDGET-MANAGER für den Schneider CPC 464 und CPC 664 kostet einschließlich umfangreichem Handbuch DM 148,–**

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,– Versandkosten
Nachnahme ☐ Verrechnungsscheck liegt bei

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

tel, die nicht jedes Forth-System zu bieten hat.
Aber auch nach dem Kauf lassen die Leute von Forth-Systems ihre Kunden nicht alleine. Neben dem Programm kann noch Fachliteratur, spezielle Forth Magazine und die Mitgliedschaft in Forth Interrest Group erworben werden. Auch bei Fragen zu dieser leistungsfähigen Sprache steht das Team gerne zur Verfügung. (TM)



Deisys

Hersteller: EDV-Beratung Worms
Vertrieb: EDV-Beratung Worms
Steuerung: Tastatur
Monitor: Farbe/Grün
Programm: 100% Basic/M-Code
Unverb. Preisempfehlung: 198,– DM
CPC 464 ☒ CPC 664 ☐ CPC 6128 ☐

Datenbanken sind heutzutage ein Haupteinsatzgebiet der Microcomputer. Dieses reicht von spezialisierten Datenbanken über generative Dateisysteme bis hin zu intelligenten Banken. In einer Datenbank legt ein User Daten nach einer bestimmten Maske ab. Bei spezialisierten Datenbanken ist diese Maske bereits definiert und auch die Auswertung dieser Daten erfolgt nach einem fest programmierten Schema.

Freie Datenbanken, die sogenannten generativen, ermöglichen es dem Anwender, seine Eingabe- und Auswertungsmasken selbst zu definieren.
Leider ist es beim CPC bisher Standard, Dateien sequentiell anzulegen, da die relative Dateiverwaltung doch ein bißchen mehr als Basic-Kenntnisse voraussetzt.
Diese Form der Abarbeitung stellt jedoch höchste Ansprüche an Struktur des Programms und Kapazität des Rechners.

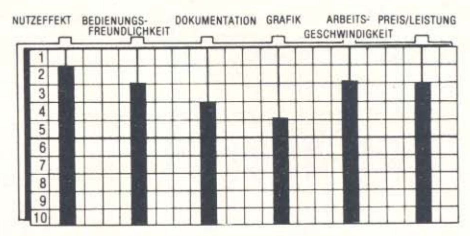
Deisys ist nun ein generatives Dateisystem, das aufgrund seiner internen Programmierung durchaus in der Lage ist, Disketten-orientierte Relativ-Dateien aufzubauen. Der Vorteil einer Relativ-Datei liegt darin, daß sich das Gross der Daten nicht im Speicher sondern auf der Diskette befindet.
Hier kann nun unter Bestimmung der internen Programmierlogik auf einzelne Diskettenabschnitte zugegriffen werden.
Dies bringt zunächst einmal eine wesentlich höhere Kapazität als mit einem 64Kb-Rechner verwirklicht werden kann, da auf der Diskette ja bekanntlich weit über 100Kb zur Verfügung stehen, zum anderen, laufen Such- und Sortierroutinen mit wesentlich höherer Geschwindigkeit als bei sequentiellen, Speicher-orientierten Dateien ab.

Bewertungsschema:
Unsere Bewertungsmatrix zeigt waagerecht die Bewertungskriterien Sound, Grafik etc... und senkrecht die Noten 1 - 10, wobei 1 für „absolut super" und 10 für „indiskutabel" steht.

Der Anwender hat bei Deisys die Möglichkeit, Bildschirmeingabefelder selbst zu bestimmen. Je nach Anwendung sind diese Masken völlig verschieden. Ein Hausanwender wird zum Beispiel ein Bemerkungsfeld mit Hobbys versehen, ein Automechaniker eher mit dem Wagentyp seines Kunden.

Generell kann nach allen Einträgen der Maske gesucht werden. Will der Automechaniker zum Beispiel alle Kunden auflisten, die als Fahrzeug einen Porsche besitzen, so gibt er den entsprechenden Index als Hauptsuchfeld an und bekommt vom System alle auf der Diskette vorhandenen Porsche-Besitzer gelistet.

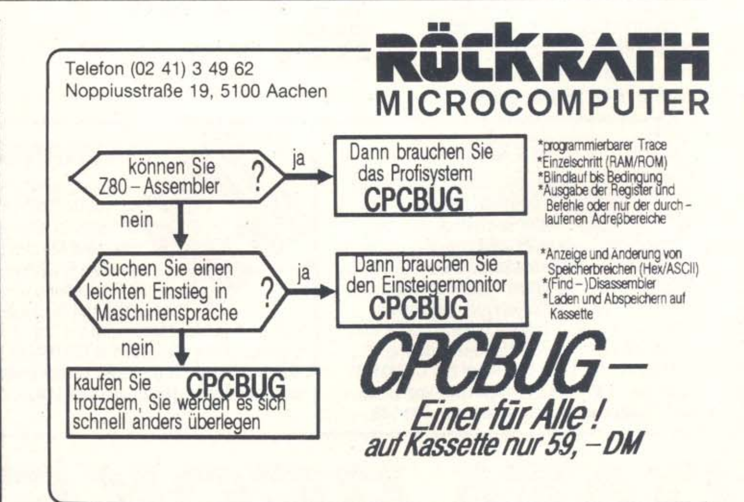
Dies geschieht mit einer Geschwindigkeit, an die normale Speicher-orientierte Dateien niemals heranreichen können, da bei Diskettenorganisation so lästige Vorgänge wie Garbagecolection entfallen.
Das Programm arbeitet also nicht spezialisiert und stellt aufgrund seiner internen Logik ein gelungenes Stück Software sowohl für professionelle als auch Hobbyanwendungen dar.
Nach dem gleichen Prinzip der relativen Dateiverwaltung bietet der Hersteller auch noch das Paket VERIS an, das eine spezialisierte Dateiverwaltung für Versicherungsagenturen darstellt. Auch hier dominieren die Abarbeitungsgeschwindigkeiten und Speicherkapazität über den Programmen, die bisher zu erhalten waren.
Da der Autor selbst im Versicherungswesen tätig ist, ist auch hier ein hoher Standard an Bedienungsfreundlichkeit für spezialisierte Anwendungsbereiche gegeben.
Einziges Manko der ganzen Angelegenheit ist, daß diese Programme nur den Benutzern einer Floppy-Station offenstehen.
Allerdings wird sich so mancher Besitzer der FD-1 wundern, zu welchen Leistungen sein Laufwerk fähig ist, wenn man es mit einem gut entwickelten Programm verbindet. (TM)



Graphicmaster

Hersteller: Vortex
Vertrieb: Vortex
Autor: Vornheim/Maier
Steuerung: Tastatur/Joystick
Monitor: Farbe/Grün
Programm: 100% M-Code
Preis: 79,– DM
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Graphicmaster ist ein Programm, mit dem sowohl der Anfänger als auch der Fortgeschrittene in der Lage ist, auf seinem Rechner eindrucksvolle Grafiken zu erstellen.
Hilfsfunktionen wie Linien ziehen,



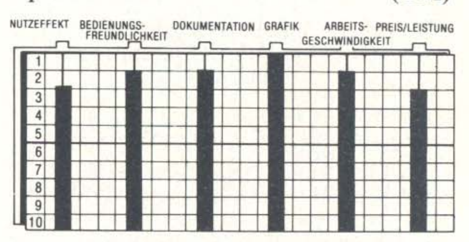
Kreise malen, beliebige Flächen mit Farbe füllen oder Einzelheiten des Bildes vergrößern und editieren, sind über Menue anwählbar.



Gearbeitet werden kann in allen drei Modi des CPC. Unterschiede bestehen lediglich in der Anzahl der Farben, die dem Anwender zur Verfügung stehen. Da zusätzlich zur Tastatur auch noch mit dem Joystick gearbeitet werden kann, geht der Umgang mit dem Zeichenstift leicht von der Hand und auch Nicht-Grafiker sind in der Lage, innerhalb von Minuten Titelbilder für ihre Programme zu entwerfen oder sich einfach nur von den bisher unentdeckten künstlerischen Fähigkeiten überraschen zu lassen. Folgende Hilfsroutinen können per Tastendruck oder Joystick angewählt werden:

MALEN
Freies Zeichnen von Linien und Punkten
LUPE
Zeichnen mit achtfacher Vergrößerung
NEU
Bild löschen
DRUCK
Ausgabe einer Hardcopy
TEXT
Einfügen von Texten
MODUS
Zeichenart wählen
DISK
Diskettenmenue
MUSTER
Abspeichern und Laden von Bildschirmteilen
ENDE
Rücksprung zu Basic
FÜLLEN
Einfärben einer beliebigen Fläche
KREIS
Zeichnen von Kreisen und Elipsen
RAHMEN
Zeichnen eines Rechtecks
STIFT
Wählen und editieren des Zeichenstifts
LINIE
Zeichnen verbundener gerader Linien
STRAHL
Zeichnen von Linien mit gleichem Anfangspunkt
KOPIE
Kopieren eines Bildteils an eine andere Stelle
BOX
Zeichnen eines ausgefüllten Rechtecks
SCHEIBE
Wie Kreis, jedoch anschließendes Füllen mit Stiftfarbe
FARBEN
Auswahl der Stiftfarbe

Diese Angaben der Leistungsdaten sprechen für sich.
Selbstverständlich können erstellte Grafiken auch vom Standardbasic aus geladen werden.
Dieses Grafiktool, das sowohl auf Vortex Floppy als auch im FD-1 Format erhältlich ist, stellt eine sinnvolle Hilfe für Programmierer von Adventures und Spielen dar. (TM)





Combat Lynx

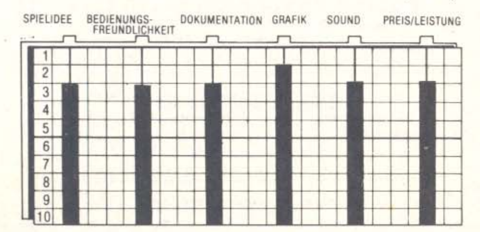
Hersteller: Durell
Vertrieb: ZS-Soft
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100% M-Code
Unverb. Preisempfehlung: 39,– DM
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Taktische Simulationen gehören zu den interessantesten Themen, die sich auf einem Computer verwirklichen lassen. Spätestens seit dem Film War-Games sind solche Programme bekannt und werden in immer größeren Mengen für die Strategen unter den Telespielern auf den Markt gebracht.
Die Begeisterung am Nachstellen solcher fiktiven Situationen fand ihren Anfang mit Zinnsoldaten und wurde im Laufe der Zeit immer mehr perfektioniert. In Amerika und England gibt es Clubs, die sich in ihrer Freizeit mit nichts anderem beschäftigen als große Schlachten auf Pappbrettern nachzuvollziehen. Etabliert in diesem Geschäft hat sich die Firma Avalon Hill, die neben den bekannten Dungeons und Dragons Brettspielen die sogenannten Hex-Games entwickelten. Bei einem solchen Spiel kontrolliert der Spieler bis zu 3000 Spielsteine auf einem riesigen Brett, das in kleine 16-eckige Felder aufgeteilt ist. Nachdem der Computer jedoch immer mehr die Haushalte eroberte, kam man auf die Idee, solche komplexen Geschehen auch auf den Bildschirm zu bannen. Innerhalb kürzester Zeit entwickelte sich ein neuer Spieltypus, die strategischen Simulationen.
Hierbei gibt es jedoch auch noch zwei Grundtypen zu unterscheiden. Das sind zum einen die War-Games, die die perfekte Umsetzung eines Hex-Games darstellen, und die War-Simulation. Die letzteren sind strategische Spiele, die dem Spieler neben taktischem Überlegen auch noch Geschick abverlangen.
Combat Lynx ist eine War-Simulation, in der ein Spieler einen Hubschrauber steuert, Angriffsziele vernichtet und die Truppen- und Materialversorgung von Stützpunkten übernimmt.
Zu diesem Zweck muß der Hubschrau-

ber, ein Westland Lynx, zunächst einmal gelandet und beladen werden. Hierbei kann der Spieler wählen, wieviel Ausrüstung und Menschen er bei einem Flug an Bord nehmen will. Begrenzt wird die Aufnahme durch die maximale Tragkraft des Hubschraubers.
Während des Ladevorgangs rotiert über der Statuszeile ein drei-dimensionales Aufrißmodell der Maschine.
Ist der Ladevorgang abgeschlossen und der Hubschrauber aufgetankt, so kann sich der Pilot in den Combat (Kampf) werfen.
Bei der unheimlich schnellen Flugphase mit eindrucksvoller 3D-Darstellung, kann der Spieler Navigationskarten einblenden und auf dem Bildschirm scrollen.



Dabei hat er eine ständige Kontrolle über das Statuswindow, in dem Meßgeräte, Kompaß und Ladekontrolle ersichtlich sind.
Ein weiteres Window an der Seite des Bildschirms gibt über die Flughöhe Auskunft. Während des Fluges muß sich der Flieger ständig Angriffen der feindlichen Streitkräfte zur Wehr setzen.
Das erfordert schnelle Reaktionen und gutes Vorausplanen.
Das Spiel ist eine gelungene Mischung zwischen Flugsimulator und Strategie-Spiel. Die fliegerischen Eigenschaften entsprechen durch Zusammenarbeit mit der technischen Abteilung der Westland Helicopter Werke, die die Lynx herstellen, der Realität und durch den gleichzeitigen Ablauf verschiedener Bewegungen in den einzelnen Bildschirmteilen kann man hier von echtem Realtime-Multitasking sprechen.
Es bleibt noch zu sagen, daß unter all diesen hervorragenden Features auch der Sound und die Grafik nicht gelitten haben und auch hier bei den Scrollbewegungen des Hintergrunds einwandfreie Arbeit geleistet worden ist.
Das Programm hat sich einen Platz an der Spitze der Simulations-Spiele wohlverdient. (TM)



Codename Mat

Hersteller: Micromega/Amsoft
Vertrieb: Data Media
Steuerung: Tastatur/Joystick
Monitor: Farbe/Grün
Programm: 100% M-Code
Unverb. Preisempfehlung: 39,– DM
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

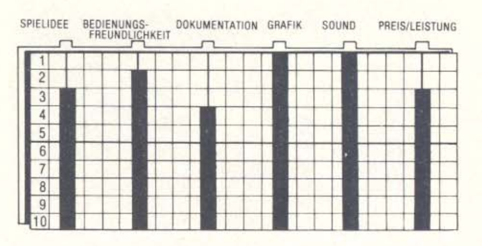
Seit Jahrhunderten versuchen die Flotten von Myon, das Solare System zu erobern. Heutzutage, in der letzten Phase des Kampfes, an dem alle anderen galaktischen Kreuzer gescheitert sind, unternimmt die Erdregierung ihren letzten Versuch, die Menschheit zu retten. Das Gehirn eines Telespielbegeisterten Teenagers wird mit den genialsten Taktikern des Universums verschmolzen, um die Menscheit zu retten. Sie übernehmen in diesem Spiel die Rolle von MAT, des Lebewesens, das aus dieser Gehirnverschmelzung hervorgegangen ist. Auf Ihrem Raumschiff, der USS Centurion, stellen Sie sich dem Angriff der feindlichen Myons.
Hierbei handelt es sich jedoch nicht nur um eines der Ballerspiele, die zur Genüge bekannt sind, hier erfordert es taktisches Denken, Kommunikation mit anderen Computern und das perfekte Zurechtfinden auf Sternenkarten.
Sie sind MAT und damit die letzte Hoffnung der Menscheit.
Das Spiel hat eine ausgezeichnete Grafik und fordert dem Rechner und dem Spieler, das Letzte an Logik ab.

Hier ist Ihnen nicht nur mit einem schnellen Joystick gedient, Sie müssen sich auf der Tastatur Ihres Rechners auskennen wie in der eigenen Hosentasche.
Im Gegensatz zu allen bisherigen Spielen aus England, ist die Anleitung ausgezeichnet und dokumentiert die Spielhandlung perfekt.
Vom einfachen Kartencomputer bis hin zum komplizierten Hypersprungzielsystem, will alles perfekt bedient werden.



Leider sind die Anleitungs- und die Programm-Dialoge in englisch gehalten. Allerdings kann von einem geschickten Taktiker erwartet werden, daß er sich auch mit der Bedienungsanleitung zurechtfindet.

Grafisch und soundmäßig bestimmt ein Spiel der oberen Spitzenklasse. Von dem Programmablauf in die Richtung Taktikspiele einzuordnen.
Insgesamt eine Kombination von einer hervorragenden Spielidee und hervorragendem, programmiererischen Können. (TM)



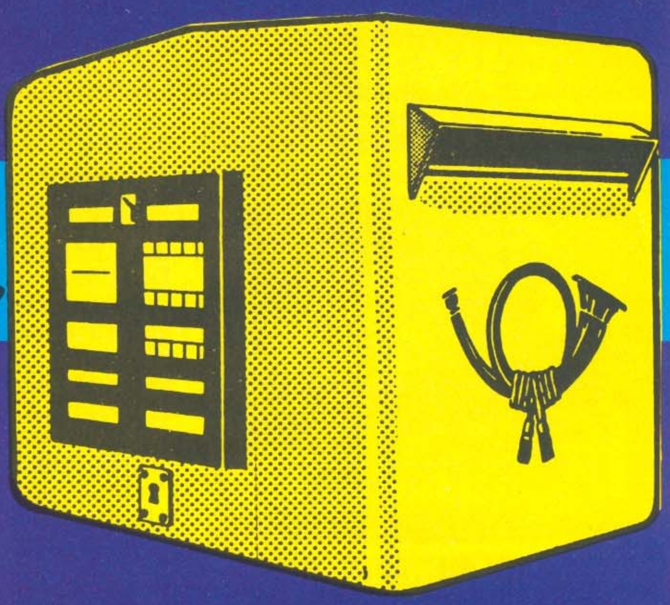
Alien 8

Hersteller: Ultimate
Vertrieb: ZS-Soft
Monitor: Farbe/Grün
Bedienung: Joystick/Tastatur
Programm: 100% M-Code
Unverb. Preisempfehlung: 39,– DM
CPC 464 ☒ CPC 664 ☐ CPC 6128 ☐

Seit undenklichen Zeiten treibt ein gewaltiges Raumschiff im Intergalaktischen Leerraum. Dunkelheit und Stille herrschen in den Räumen und Korridoren. Im freien Fall sind alle Schiffssysteme inaktiv, nur der große Bordcomputer hält seit langer Zeit Ausschau nach dem Zielplaneten der Reise. Und dann ist es soweit. Vor dem Raumschiff taucht ein Sonnensystem aus dem Dunkel des Weltalls auf. Der Computer fährt alle Systeme auf Normalleistung hoch. Als dies alles geschehen ist, wird ein Androide aktiviert, dessen Aufgabe es ist, die schlafende Besatzung zu revitalisieren: ALIEN 8.
Wie schon bei allen anderen Ultimate Programmen wurde auch Alien 8 in eine phantasievolle Rahmenhandlung eingebettet. Wie bereits aus dieser Geschichte zu ersehen ist, haben Sie die Aufgabe, den Androiden Alien 8 zu steuern und die Gegenstände, die für die Wiederbelebung der Besatzung benötigt werden, einzusammeln und an den Ort zu bringen, an dem sie gebraucht werden. Also, so werden Sie nun sagen, eines der altbekannten Jump & Run Spiele – aber was für ein Spiel. Der wesentlichste Unterschied zu Spielen à la Manic Miner und Co. ist das völlig neuartige Bildschirmdisplay. Kannte man bisher nur einen zweidimensionalen Screeninhalt der bei dem einem oder anderen Programm mit einem quasi 3-D Effekt garniert wurde, wird man bei Alien 8 nicht nur mit einem grafischen Schnörkel konfrontiert. Vielmehr ist das gesamte Spielgeschehen drei-dimensional aufgebaut. Es ist tatsächlich möglich seine Spielfigur im Bildschirmszenario vor, zurück, links, rechts, auf und ab zu bewegen. Ebenfalls besteht die Möglichkeit, bestimmte bewegliche Gegenstände umherzuschieben, um sie als Brücke oder Leiter zu verwenden. Man kann ruhigen Gewissens behaupten, daß Alien 8 grafisch zum Besten zählt, was es für den CPC momentan gibt.

data media

MAILORDER



Wie bestellen!
Die Bestellkarte im Innenteil der CPC heraustrennen, ausfüllen und an DATA MEDIA senden. Lieferung erfolgt umgehend! Zahlung per Vorkasse oder Nachnahme zzgl. Porto- bzw. Nachnahmegebühr. (Nachnahme ins Ausland ist nicht möglich.)

Anwender-Software

Anwenderprogramme für CPC 464/664

Focus *UNSER RENNER!*
Programm zur perspektivischen Darstellung von 3D-Grafiken. Stellen Sie z.B. ein Haus dar und verändern Sie durch Eingabe der Winkel die Perspektive!
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Sekretariat
Ein Programmpaket bestehend aus: Textverarbeitung, Adressverwaltung und Faktura.
Cassette **179,00 DM** Diskette 3", 5 1/4" **189,00 DM**

Terminplaner
Planen Sie Ihre Termine! Wochen- und Monatsübersicht.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

Turbo Tape
Das Programm erhöht die Datenausgabe zwischen Rechner und Kassette erheblich.
Cassette **29,00 DM**

Neu **Multidatei**
Ein universelles Dateiverwaltungsprogramm.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

Zeichengenerator
Hilfsprogramm zum bildschirmorientierten Definieren von eigenen Zeichen. Laden, Sichern und Ausgeben von definierten Zeichen. Besonders geeignet für Adventures.
Cassette **49,00 DM** Diskette 3", 5 1/4" **59,00 DM**

Routinen, Tips und Tricks
Ca. 50 wichtige Programmroutinen für jeden Programmierer, wie z.B. Sortierroutine, Inkeyroutine, Diskettenroutine usw.
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Textverarbeitung
Aufwendige Textverarbeitung mit deutschem Zeichensatz und mathematischen Sonderzeichen. Schnittstelle zur Data Media Adressverwaltung vorhanden.
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Adressverwaltung
Kommerzielle Adressverwaltung zur Verarbeitung von Adressen. Schnittstelle zur Data Media Faktura und Data Media Textverarbeitung.
Cassette **67,00 DM** Diskette 3", 5 1/4" **79,00 DM**

Faktura
Schreibt Rechnungen, Mahnungen, Lieferscheine, Bestellungen, Quittungen und kalkuliert das Vertriebsprogramm und Verkaufspreise.
Diskette 3", 5 1/4" **89,00 DM**

Kassenbuch
Tageskassenbericht mit Einnahmen und Ausgaben, Kassenbestand bei Geschäftsschluß, Entnahmen sowie Ausgabe von Buchungsbelegen.
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Neu **Basic Compiler**
Übersetzt Ihre Basicprogramme in Maschinensprache und erhöht dadurch die Verarbeitungsgeschwindigkeit
Cassette **89,00 DM** Diskette 3", 5 1/4" **99,00 DM**

Vereinsverwaltung
Programm für Vereine zur Erleichterung der Verwaltung von Mitgliedern, Beitragszahlungen, Jubiläen, Geburtstage, Statistiken, Buchhaltung usw.
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Verwaltungsarchiv
Verwaltungsprogramm für Telefonnummern, Archiv für Bücher, Videocassetten, Schallplatten, Musikcassetten und Disketten.
Cassette **69,00 DM** Diskette 3", 5 1/4" **79,00 DM**

Lagerverwaltung
Lagerverwaltungsprogramm für jeden Betrieb. Automatische Artikelüberwachung, Aktualisierung der auf Datenträger gespeicherten Daten und Ausdruck von Bestands-, Bestell- und Verkaufslisten.
Cassette **79,00 DM** Diskette 3", 5 1/4" **89,00 DM**

Neu **Platinenkit**
Erstellt auf einfache Weise Layouts Ihrer elektronischen Schaltungen.
Ausgabe auf Drucker
Cassette **189,00 DM** Diskette 3", 5 1/4" **199,00 DM**

Neu **Gredi**
Hilfsprogramm zur Herstellung von Hires-Grafiken
Diskette 3", 5 1/4" **59,00 DM**

Neu **Abersoft Forth**
Programmiersprache Forth
Diskette 3" **148,00 DM**

Reisekosten
Eingabe aller Abrechnungsbelege, Personalnummer, Reisekostenvorschuß usw. Erstellung einer kompletten Reisekostenabrechnung für In- und Ausland unter Berücksichtigung der üblichen Tages- und Kilometerpauschalen.
Cassette **69,00 DM** Diskette 3", 5 1/4" **79,00 DM**

Videothek
Archivprogramm für Video und Film. Die Angaben umfassen: Titel, Spielzeit, Rangfolge, Regie, Darsteller usw.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

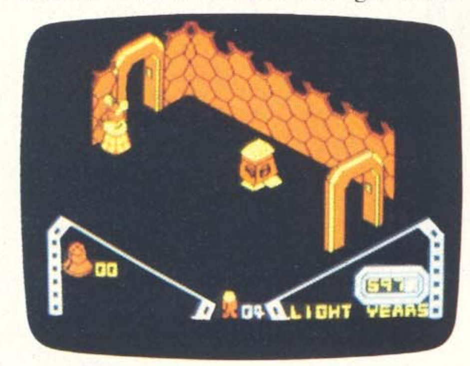
Münzarchiv
Archivprogramm für Münzen. Die Angaben umfassen: Katalog-Nr., Ordnungskriterien, Motive, Katalogpreise, Anzahl, Alter usw.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

Briefmarkenarchiv
Archivprogramm für Briefmarken. Die Angaben umfassen: Katalog-Nr., Ordnungskriterien, Motive, Katalogpreise, Anzahl usw.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

Bibliothek
Archivprogramm für Bücher. Die Angaben umfassen: Titel, Verlag, Autor, Auflage, Sachbücher, Fachbücher, Belletristik usw.
Cassette **59,00 DM** Diskette 3", 5 1/4" **69,00 DM**

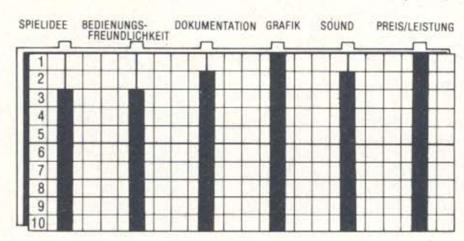
Weitere Artikel in unserem Gesamtkatalog. Bitte anfordern (2,– DM für Rückporto beilegen.)
DATA MEDIA GmbH, Bereich Mailorder, Postfach 1263, 4620 Castrop-Rauxel Telefonische Auskünfte: 0231/12 50 71
In der Schweiz: Data Media Schweiz, Bungertstraße 15, 8802 Zürich-Kilchberg, Tel.: (01) 715 36 19

Alle Gegenstände, Räume bis hin zu den schlafenden Besatzungsmitgliedern wurden von den Ultimate Programmierern mit Fingerspitzengefühl entworfen, und mit viel Liebe, zum Detail, auf den Bildschirm gebracht.



Aber nicht nur diese Rahmen-Komponenten können sich sehen lassen, denn auch der Spielablauf ist dafür, daß es ja nicht die neueste Spielidee ist, hervorragend gemacht. Die Revitalisierung der Besatzung erfolgt über sogenannte Aktivatoren, eine Maschine, die in jeder Schlafkammer steht und die Besatzung in Tiefschlaf versetzt, sie währenddessen am Leben erhält, und außerdem dazu da ist, die Schläfer wieder zu wecken. Nun ist auf der langen, gefahrvollen Reise nicht alles so gelaufen wie es eigentlich sollte. Die Aktivatoren sind nicht mehr in der Lage den Wiedererweckungs-Prozeß alleine einzuleiten – sie sind defekt. Der Androide Alien 8 muß nun im gesamten Schiff die Ersatzteile, welche für die Reparatur der im Schiff befindlichen 24 Aktivatoren benötigt werden, einsammeln und diese reparieren.
Dies erweist sich als ein echtes Problem, denn in einem derart langen Zeitraum haben sich im Schiff Wesen entwickelt, die für einen kleinen Androiden zu einer echten Gefahr werden können. So bevölkern nun aufziehbare Mechanomäuse, schillernde Energiewesen und aufrecht gehende Monstermäuse das Schiff. Auch sind viele Gegenstände des Schiffs-Inventars zu Todesfallen geworden. Treppen und Blöcke, die sich bewegen oder verschwinden, wenn man auf sie steigt, mörderische Pyramiden deren Berührung jedesmal mit dem Verlust eines Lebens enden, und Minen, die einen ebenso fatalen Effekt erzielen. Aber glücklicherweise gibt es immer noch Objekte an Bord, die auch hilfreich sein können. Das interessanteste Hilfsmittel ist zweifelsohne eine Art Robodroid, der über eine Steuerkonsole gelenkt werden kann und sich hervorragend dazu eignet, Minen aus dem Weg zu räumen.

Auch auf eine andere Weise unterscheidet sich Alien 8 von anderen Spielprogrammen; es werden keine Punkte mehr vergeben. Vielmehr bekommen Sie am Ende eines jeden Spieles eine Wertung über Ihre Qualitäten als Spieler. Die Bewertungsskala reicht von 'armselig' bis 'absoluter Abenteurer'. Außerdem wird Ihnen noch angezeigt, wieviele Kammern Sie jeweils aktiviert haben.
Einen Wermutstropfen gibt es allerdings in diesem hervorragenden Programm: Es mangelt an einer deutschen Anleitung. Die originale Anleitung, die jeder Packung beiliegt, ist in einem nur schwer verständlichem Englisch gehalten.
Doch auch dieses Handicap läßt sich vertreten, denn bereits nach einigen Spielen ist man in der Lage sich zusammenzureimen, was man nun eigentlich tun muß. Alien 8 gehört zu der Art von Programmen, die man getrost auf eine einsame Insel mitnehmen kann und dabei die Sicherheit hat, lange Freude daran zu haben. (HS)



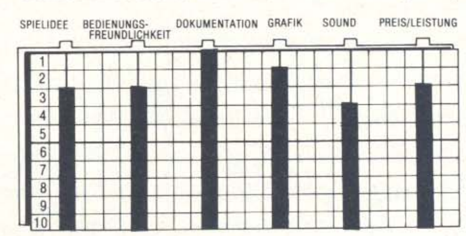
Roland goes Square Bashing

Hersteller: Durell/Amsoft
Vertrieb: Data Media
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: 39,– DM
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Roland goes Square Bashing ist eine geschickte Mischung zwischen Labyrinth-Spiel und dem bekannten Q-BERT.



Hier gilt es nur eine scharfe Kombinationsgabe und ein drei-dimensionales Denken zu haben. Angegriffen von schleimigen Monstern oder Killer Gorillas wird keiner. Dafür wurde Wert auf optisch einwandfrei und komplizierte Labyrinthe gelegt. Der Spieler Roland muß über 30 verschiedene Labyrinthe, die in Form von gestaffelten Platten im Raum liegen, hinwegkommen. Dazu muß erst einmal das Labyrinth, das mit den Effekten der optischen Täuschung aufgebaut ist, übersehen und der richtige Weg auskalkuliert werden. Allerdings muß man sich dabei ganz schön beeilen, da die Platten, auf denen der Spieler steht, immer kleiner werden, bis sie verschwinden und der Spieler im Hyperraum verlorengeht. Im Gegensatz zu den normalen Labyrinth-Spielen ist der Spieler hierbei nicht gezwungen, die Irrgärten der Reihe nach durchzuarbeiten, sondern kann frei bestimmen, welchen der vielen Level er als nächsten bestehen möchte.
Durch den Lerneffekt, der dem Spiel innewohnt und die unterhaltsame Grafik sowie das friedliche Spielthema, ist das Spiel auch für Kinder geeignet. Aber auch ältere Spieler werden hieran schnell Gefallen finden. (TM)

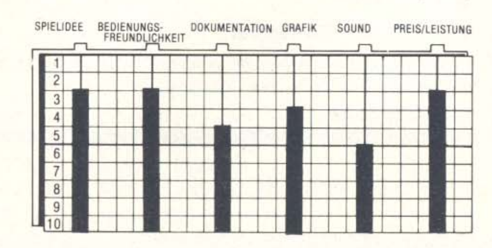


Bridge-It

Hersteller: Epicsoft/Amsoft
Vertrieb: Data Media
Autor: Said a Abouelhassan
Steuerung: Joystick/Tastatur
Monitor: Farbe
Programm: 100% M-Code
Preis: 39,– DM
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

In Bridge-It spielen Sie einen Klappbrückenwärter, der in einem kleinen Dorf seinen Dienst versieht, in dem alle Häuser durch Wasserstraßen voneinander getrennt sind.
Diese Brücken sind normalerweise hochgezogen, so daß Schiffe die Flüsse befahren können. Wollen sich die Bewohner des Dorfes nun untereinander besuchen, so müssen Sie im richtigen Augenblick die entsprechende Klappbrücke herunterlassen. Gelingt Ihnen das nicht, so nehmen die Fußgänger ein unfreiwilliges Bad.
Von Ihrem Wärterhäuschen aus, steuern Sie vier Brücken. Gute Beobachtungsgabe und eine schnelle Reaktion sind Voraussetzungen, die Sie in Ihrem Job mitbringen müssen. Durch eine schräge Draufsicht auf das Dorf hat das Spiel einen schönen 3D-Effekt. Die Soundeffekte sind zwar eher als mäßig zu bezeichnen, haben jedoch auch den Vorteil, den Spieler nicht abzulenken.

Kindern kann dieses Spiel geruhsam in die Hand gegeben werden, da ein gewisser Lerneffekt und die unterhaltsame Grafik sowie das friedliche Spielthema für die niedrigen Altersgruppen ideal sind. (TM)



Sound mit dem CPC Teil 7

Erinnern Sie sich noch? In unserem vergangenen Sound-Kurs hatten wir unsere Orgel schon soweit ausgebaut, daß per Tastendruck die Oktaven geändert werden konnten und die Töne zu mehreren Kanälen geschickt wurden. Wie versprochen, werden wir heute diese Orgel weiter ausbauen und verfeinern, so daß wir das Erlernte gleich in eigene Programme einbinden können.

Bevor wir nun an die Arbeit gehen, müssen zunächst die Funktionen definiert werden, die unsere Orgel später erfüllen soll. Als Erweiterungen stehen diesmal an:

- Die über die Tastatur gespielten Töne sollen so aufbereitet werden, daß sie später per Tastendruck als Data-Zeilen ausgegeben werden können. Als Ausgabemedium soll entweder der Bildschirm oder der Drucker dienen.
- Ein sogenannter Playback-Mode soll eingebaut werden, der auf Tastendruck die gespielte Melodie in verschiedenen Geschwindigkeiten nachspielt.

Zunächst müssen wir für die Perioden- und Oktavenwerte ein Feld dimensionieren, das die Anzahl der zu spielenden Töne festhält. Diese Werte haben wir in Zeile 130 definiert, die höchstmögliche Anzahl ist also 500 Töne. Bei Bedarf können Sie diese Werte selbstverständlich beliebig ändern. Daraus folgt dann auch die Veränderung der Zeilen 160-220, in denen der Zähler jeweils bei einer gespielten Note um 1 erhöht wird.

Anschließend müssen wir eine Unterroutine schreiben, in der die Data-Werte berechnet und gespeichert werden, um sie dann auf Tastendruck abrufen zu können. Diese Routine beginnt mit Zeile 290 zunächst mit der simplen Abfrage des Ausgabegerätes, also Drucker oder Bildschirm. Wie Sie in Zeile 310 erkennen können, ist die Geräteadresse des Druckers 8. Für den Bildschirm brauchen wir keine Gerätenummer zu definieren, da automatisch alle Ausgaben zunächst auf den Schirm gebracht werden. Eine Besonderheit ist noch, daß wir die Ausgabe der Data-Zeilen beeinflussen können. Die Zeilen 310-320 fragen die Zeilennummer und die gewünschte Schrittweite ab, mit denen die Datas dann ausgelesen werden. So erspart man sich beim Einbau der Datas in eigene Programme das spätere Umnumerieren der entsprechenden Zeilen. Das eigentliche Auslesen der Werte folgt in Zeile 350, wobei a die Periode und b die Oktave ausliest. Der Aufruf für das Lesen dieser Werte geschieht in Zeile 250, beim Drücken der Pfeil-Taste wird das Unterprogramm angesprungen.

Die nächste Subroutine beginnt ab Zeile 390 und enthält das sogenannte Playback, also die Wiederholung der gespielten Melodie. Da wir die Melodie ja in verschiedenen Geschwindigkeiten wiederholen möchten, müssen wir eine Abfrage der Verzögerung (Zeile 400) einbauen. Als sinnvoll erschien uns eine Verzögerung zwischen 1-30, wobei 30 etwa der Originalgeschwindigkeit entspricht. Diese Werte können allerdings beliebig geändert werden. – Experimentieren hilft auch hier über vieles hinweg.

Im großen und ganzen waren das bereits die benötigten Routinen, um unsere ehemals kleine Orgel zu einem komfortablen Instrument auszubauen. Das Unterprogramm „Playback" wird in Zeile 260 mit Druck auf die „-"-Taste aufgerufen.

Was noch bleibt, ist die Erklärung des Warm- bzw. Kaltstarts in Zeile 260. Diese Abfrage ermöglicht es, nach einer gespielten Melodie mit den gespeicherten Werten fortzufahren – sie bleiben also erhalten, wenn Sie einen Warmstart durchführen. Bei einem Kaltstart werden alle Werte gelöscht und das Programm startet von vorn.

Allerdings haben wir auf die Möglichkeit des Mehrklanges verzichtet, um die Übersichtlichkeit zu bewahren. In unseren nächsten Sound-Lehrgängen werden wir jedoch diese Schritte erarbeiten und in unseren z.Zt. noch „monophonen Synthesizer" einbinden.



Übrigens würden wir uns über Anregungen, Verbesserungen sowie Kritiken sehr freuen, da gerade dieses Kapitel eine Fülle von ungeahnten Möglichkeiten bietet. Neben dem Mehrklang wird unser großes Ziel der nächsten Lehrgänge auch die grafische Darstellung von Noten sein. Doch bis dahin ist noch ein weiter Weg, den wir uns gemeinsam erarbeiten werden. Wir wünschen zunächst viel Spaß beim Experimentieren und würden uns über Zusendung selbsterstellter Sound-Programme freuen. Als Zugabe gibt es wie immer ein Beispielprogramm, das uns diesmal als sehr gelungen scheint. (SR)

```
10 REM *********************************
*
20 REM *   monophoner      synthesizer
*
30 REM *     mit playback und datas
*
40 REM *  ---------------------------
*
50 REM * bedienung q-u : toene erzeugen
*
60 REM * bedienung '↑' : datas erzeugen
*
70 REM * bedienung '-' : playback mode
*
80 REM *  ---------------------------
*
90 REM *           by werner 1985
*
100 REM ********************************
**
110 MODE 2
120 CLS:PRINT"                S   Y   N   T
   H   E   S   I   Z   E   R":PRINT"----
---------------------------- ACTIVATED
---------------------------"
130 DIM a(500): DIM b(500)
140 oktave =2
150 a$=INKEY$
160 IF a$="q"THEN periode=119:a(b)=perio
de:b(b)=oktave:b=b+1:GOTO 280
170 IF a$="w"THEN periode=106:a(b)=perio
de:b(b)=oktave:b=b+1:GOTO 280
180 IF a$="e"THEN periode=95:a(b)=period
e:b(b)=oktave:b=b+1:GOTO 280
190 IF a$="r"THEN periode=89:a(b)=period
e:b(b)=oktave:b=b+1:GOTO 280
200 IF a$="t"THEN periode=80:a(b)=period
e:b(b)=oktave:b=b+1:GOTO 280
210 IF a$="y"THEN periode=71:a(b)=period
e:b(b)=oktave:b=b+1:GOTO 280
220 IF a$="u"THEN periode=63:a(b)=period
e:b(b)=oktave:b=b+1:GOTO 280
230 IF A$="["THEN oktave=oktave/2
240 IF a$="]"THEN oktave=oktave*2
250 IF a$="↑" THEN GOSUB 290:CLS:PRINT"
             S   Y   N   T   H   E   S
I   Z   E   R":PRINT"--------------------
-------------- ACTIVATED ---------------
-------------":INPUT"{W}ARM ODER {K}ALTSA
RT";X$:IF X$="k" THEN RUN
260 IF a$="-" THEN GOSUB 390:CLS:PRINT"
             S   Y   N   T   H   E   S
I   Z   E   R":PRINT"--------------------
-------------- ACTIVATED ---------------
-------------":INPUT"{W}ARM ODER {K}ALTSA
RT";X$:IF X$="k" THEN RUN
270 GOTO 150
280 SOUND 1,periode*oktave:GOTO 150
290 CLS:PRINT"datas fuer musik"
300 PRINT"------------------":PRINT:INPUT"
ausgabe {d}rucker {s}creen";device$
310 IF device$="d" THEN dev=8
320 INPUT"beginn bei zeile :";zei
330 INPUT"schrittweite     :";ste
340 FOR dat= 0 TO b
350 PRINT#dev,zei;" data ";a(dat);","b(d
at)
360 zei=zei+ste
370 NEXT dat
380 PRINT"-------":PRINT"fertig":PRINT:PR
INT"taste betaetigen":CALL &BB06:dev=0:R
ETURN
390 CLS:PRINT"playback":PRINT"---------":
PRINT
400 INPUT"Verzoegerung {1-30}";delay
410 FOR t= 0 TO b
420 SOUND 1,a(t)*b(t)
430 FOR pause= 1 TO delay*10:NEXT pause
440 NEXT t
450 PRINT"------":PRINT"fertig":PRINT:PR
INT"taste betaetigen":CALL &BB06:RETURN
```

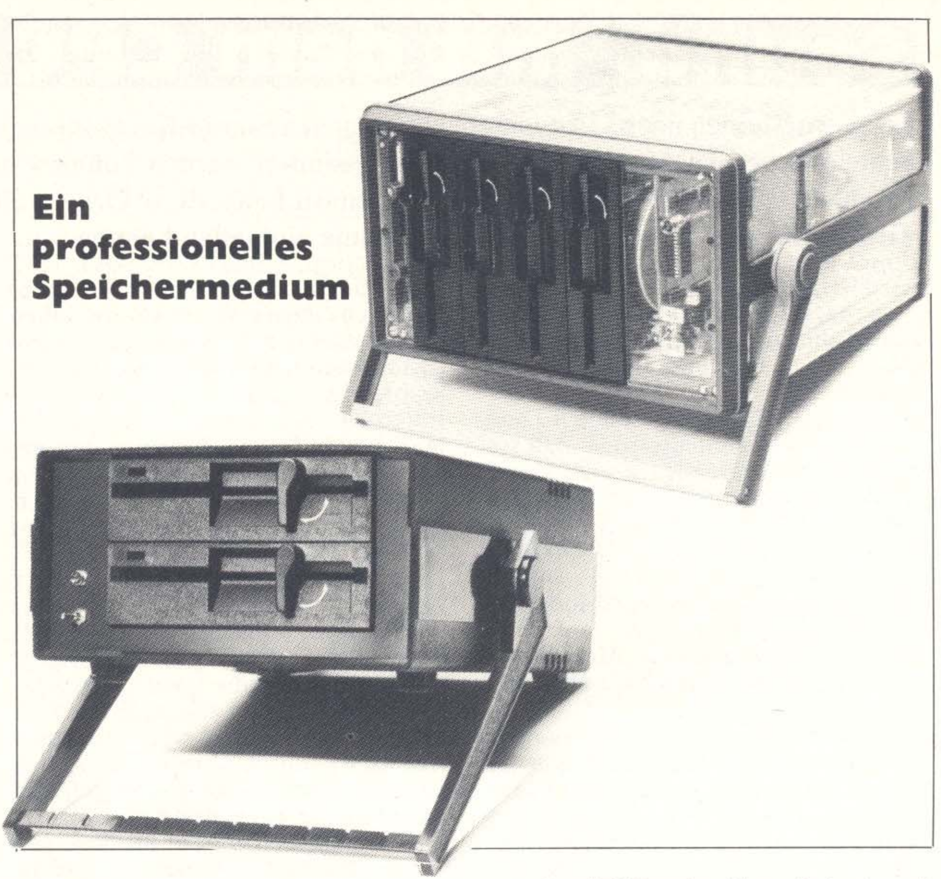
Über das Diskettenlaufwerk von Data Media gab es bisher nur Gerüchte. Die tatsächlichen Qualitäten, die nur den Insidern bekannt waren, sollen nun vorgestellt werden. Der Schneider CPC 464 und seine Nachfolger zählen zu den meistverkauften Homecomputern auf dem derzeitigen Markt. Sein günstiger Preis und seine Vorteile in Hard- und Software gegenüber den Konkurrenten haben zu dieser raschen Verbreitung beigetragen. Nachdem der Schneider Computer seine Einführungsphase erfolgreich überstanden hat, fordern die Anwender Erweiterungen, die ihnen mehr Möglichkeiten mit ihrem CPC eröffnen.
Das Diskettenlaufwerk des CPC stellt zwar eine Verbesserung gegenüber dem Kassettenrecorder dar, zeichnet sich aber mit seinen 180 KByte nicht gerade durch große Speicherkapazität aus. Dazu kommt das ungewöhnliche Format der Datenträger und der noch immer relativ hohe Preis für eine 3"-Diskette. Derjenige, der größere Programme schreibt, weiß wie schnell er mit 180 KByte am Ende ist, wenn er nicht nur sein Programm, sondern auch die dazu erforderlichen Hilfsmittel auf der Diskette unterbringen muß. Der Umgang mit diesem Diskettenformat ist dann ein gutes Training für Discjockeys. Der nächste Schritt wäre eine Festplatte, diese ist aber für die meisten Anwender zu teuer und auch noch nicht erhältlich. Es existieren aber bereits Diskettenlaufwerke mit größerer Speicherkapazität auf der Basis der 5 1/4"-Disketten.
Welche Möglichkeiten Ihnen der Schneider Computer bieten kann, soll Ihnen das Diskettenlaufwerk FDD von Data Media aufzeigen.
Die Entwicklung dieser Diskettenstation hatte zum Ziel, mit dem Schneider CPC einen großen Schritt nach vorne zu vollziehen und maximale Möglichkeiten im Rahmen dieser Hardwareerweiterung zu bieten.

Dazu gehören folgende Eigenschaften:

- möglichst große externe Speicherkapazität
- schneller Dateizugriff
- Nutzung von kommerzieller Software
- Zugriff auf gute Software für verschiedene Anwendungen
- Verarbeitung unterschiedlicher Diskettenformate
- wichtige Hardwareeinrichtungen (z.B. serielle Schnittstelle)
- und ein Maschinensprache-Monitor für die Programmentwicklung

Auf dem FDD-Laufwerk können zwei Betriebssysteme gefahren werden, FDOS (unter Schneider-Basic) und CP/M. Dabei stehen bei beiden Systemen mindestens 780 KByte Speicherkapazität zur Verfügung. Obwohl beide Betriebssysteme eigene Diskettenorganisationen besitzen, können Daten- und Textdateien mit Hilfe von Dienstprogrammen untereinander ausgetauscht werden.

Ein professionelles Speichermedium



Das Betriebssystem FDOS 1.0

Das Betriebssystem FDOS 1.0 unterstützt das Schneider-Basic und bietet eine Alternative zum Kassettenbetriebssystem und AMSDOS des Schneider Computers.

FDOS 1.0 besitzt eine ähnliche Struktur, wie sie von anderen Rechnersystemen mit DOS bekannt ist, und stellt gegenüber dem CP/M ein eigenständiges Betriebssystem mit einer anderen Diskettenorganisation dar. FDOS 1.0 besitzt eigene Befehle, die als Erweiterung des BASIC-Befehlssatzes ausgelegt sind. Das bedeutet, daß alle Kassettenbefehle erhalten bleiben und nicht vom FDOS verändert werden. Die Unterscheidung zwischen Kassetten- und Diskettenbefehlen zeigt der vorangestellte senkrechte Strich an.
Beispiel:

Load"Name" - dieser Befehl arbeitet mit der Kassette

|Load"Name" - dieser Befehl arbeitet mit der Diskette.

Unter FDOS stehen 780 KByte Diskettenkapazität für Daten und Programme zur Verfügung. Eine Datei kann unter FDOS bis zu 48 KByte groß sein. Dateinamen können gegenüber dem Kassettensystem und dem AMSDOS bis zu 24 Zeichen enthalten.

Durch Einfügen von Punkten können die Dateien mit sinnvollen Namenskombinationen bezeichnet werden. Im Inhaltsverzeichnis haben maximal 72 Dateien Platz. Beim Anzeigen des Inhaltsverzeichnisses werden maximal 20 Zeilen gelistet. Danach kann durch Betätigen der ESC-Taste die Anzeige abgebrochen oder durch eine beliebige andere Taste der Rest des Directory angezeigt werden.

Beispiel für die Ausgabe eines Disketten-Inhaltsverzeichnis:

```
|DIR
DOS 1.0 Freie Disc-Kapazität A:          710 K
BAS TEXTEDITOR                      10355 BYTE
BAS BIORHYTMUS:NEU                  16566 BYTE
BAS START.MONITOR.VERSION1             88 BYTE
BIN MONITOR.VERSION1                 8104 BYTE
BAS KALKULATION.1985                21734 BYTE
DAT FINANZEN.1985.EINNAHMEN          4011 BYTE
DAT FINANZEN.1985.AUSGABEN           2974 BYTE
BAS*SPIEL.FUER.ZWEI                  7381 BYTE
???*PROTECT.DEMO                       58 BYTE

Ready
```

Es können drei Dateitypen (BAS,BIN und DAT) verwendet werden. Erscheinen vor dem Dateinamen drei Fragezeichen, so liegt ein unbekannter Dateityp vor. Der Stern vor einem Dateinamen kennzeichnet ein geschütztes Programm.

Befehle des FDOS 1.0

Befehl	Funktion
|MON	Maschinensprache-Monitor aufrufen
|CPM	CP/M oder eine andere Programmdiskette booten
|A	Laufwerk A anwählen
|B	Laufwerk B anwählen
|C	Laufwerk C anwählen
|D	Laufwerk D anwählen
|LOAD	Programm von Diskette laden
|SAVE	Programm auf Diskette ablegen
|DIR	Inhaltsverzeichnis anzeigen
|ERA	Datei löschen
|REN	Datei umbenennen
|OPENIN	Inputdatei öffnen
|OPENOUT	Outputdatei öffnen
|CLOSE	Datei (Input/Output) schließen
|READ	Lesen aus Datei

IWRITE	Schreiben in Datei
IRUN	Programm laden und starten
IMERGE	Programme zusammenfügen
IINIT	Diskette formatieren

Alle FDOS-Funktionen (siehe Befehle) können über eine Einsprungadresse aufgerufen werden (die Funktionsnummer und die Parameter werden in Register übergeben). Dadurch können die Diskettenbefehle leicht mit Hilfe des Maschinensprache-Monitors in eigene Assemblerprogramme eingebunden werden. Zusätzlich besteht die Möglichkeit, über eine bestimmte Adresse direkte Diskettenzugriffe (Lesen/Schreiben von Sektoren) durchzuführen.

Betriebssystem CP/M

Das Diskettenlaufwerk FDD von Data Media besitzt außer dem FDOS ein Standard CP/M-Betriebssystem. Standard bedeutet in diesem Fall, daß der Schneider Computer unter CP/M die gleichen Softwareeigenschaften und den gleichen Aufbau besitzt wie andere Computer, die das CP/M Betriebssystem beinhalten.
Die Diskettenlaufwerke des CPC 464 und 664 (3") besitzen leider einen Mangel. Durch das (besonders für CP/M) ungewöhnliche Diskettenformat kann andere CP/M-Software nicht ohne spezielle Eingriffe in die Hardware genutzt werden. Das hier vorgestellte Diskettenlaufwerk beseitigt dieses Problem und eröffnet dem Schneider Computer auch ohne große Mühen die CP/M-Welt. Mit diesem Laufwerk können Disketten von anderen CP/M-Rechnern direkt gelesen und beschrieben werden.
Das BIOS (Basic-Input-Output-System) befindet sich im EPROM des Laufwerks und ist so ausgelegt, das mit wenigen Änderungen von Tabellen die meisten Diskettenformate an das vorhandene Betriebssystem angepaßt werden können.
In dem mitgelieferten Handbuch ist eine ausführliche Beschreibung mit Beispielen der Formatanpassung enthalten. Diese Formatanpassungen können durch ein Installationsprogramm in den Bootsektor eingetragen werden, so daß beim nächsten Start des CP/M-Betriebssystems dieses Format bereits eingestellt ist. Bei dem Betrieb von vier Laufwerken können vier verschiedene Diskettenformate und deren Software gleichzeitig gefahren werden.

Folgende Parameter können für die unterschiedlichen Formate geändert werden:
- Single-Double-Density
- Einseitige oder zweiseitige Disketten
- Sektorgröße von 128 Byte bis 1024 Byte pro Sektor
- Skewfaktor (Sektorversatz in einer Spur)
- Anzahl der Sektoren pro Spur
- Anzahl der Spuren von 1 x 35 bis 2 x 80
- Blockgröße
- Diskettenkapazität
- Erste Sektornummer der Spur

Alle diese Parameter sind im Bootsektor für die vier möglichen Laufwerke auf das Format der FDD CP/M-Diskette eingestellt.

Technische Daten und Merkmale

1. Controller-Karte:
Auf der Laufwerksplatine, die sich wie auch das Netzteil im Laufwerksgehäuse befindet, steckt ein Controller vom Typ WD2797. In einem 8 KByte EPROM befindet sich neben dem FDOS das Bios für CP/M und der Maschinensprache-Monitor mit den Routinen für die serielle Schnittstelle. Die Platine und das Netzteil sind für die Nachrüstung der RS 232 Schnittstelle vorbereitet und können (wenn nicht schon eingebaut) durch Einsetzen der Bauteile und der Buchse leicht mit diesem Interface ausgerüstet werden.

2. Laufwerkseigenschaften:
Die Doppel-Floppy stellt eine Diskettenkapazität von 1.6 MByte zur Verfügung, die bei Verwendung von vier Laufwerken auf maximal 3.2 MByte aufgerüstet werden kann. Die Laufwerke sind Slimline-Typen mit 2 mal 80 Tracks, die durch einen Schalter in der Gehäusefront hardwaremäßig auf 2 mal 40 oder bei einseitigem Diskettenformat auf 1 mal 40 Tracks umgeschaltet werden können. Die Steprate des Schreib/Lesekopfes ist auf den maximalen Wert von 3ms eingestellt und kann per Software geändert werden.

3. Serielle Schnittstelle
(gegen Aufpreis oder nachrüstbar)
Die RS 232 Standard Schnittstelle ermöglicht den Rechner-Rechner-Betrieb und ist für die Benutzung eines Modems voll duplexfähig. Alle Parameter sind über Software einstellbar und erlauben eine Übertragungsrate von 50 bis 19200 Baud.

4. Maschinensprache-Monitor
Der Monitor ist im EPROM untergebracht und kann nach dem Einschalten des CPC mit einem Befehl aufgerufen werden. Er beinhaltet in dieser Version alle wichtigen Befehle, um Programme in Maschinensprache zu erstellen. Mit dem Maschinensprache-Monitor hat man die Möglichkeit, Sektoren von beliebigen Diskettenformaten zu lesen und zu beschreiben oder defekte Disketten (verlorene Daten) zu reparieren. Ein erweiterter Monitor (6KByte) mit Disassembler, Tracer und weiteren Befehlen ist erhältlich.

Befehlsumfang des eingebauten Monitors:
- Booten der Diskette im Laufwerk A
- Verändern von Speicherinhalten
- Anzeigen von Speicherbereichen
- Starten von Programmen und Setzen von Breakpoints
- Lesen und Schreiben von/in I/O-Ports
- Füllen von Speicherbereichen mit Konstanten
- Verschieben von Speicherbereichen
- Ausgabe wahlweise auf Bildschirm oder Drucker
- Lesen und Schreiben der meisten Diskettenformate
- Anzeigen und Ändern der Z80-Register

Fazit:
Mit dem FDD-Laufwerk von Data Media liegt ein ausgereiftes Produkt der Spitzenklasse vor, das den Einsatzbereich des Schneider Computers nach oben abrundet. Ein für den professionellen Anwender ausgerichtetes Speichermedium, das neben den exzellenten Leistungsdaten vor allem durch seinen günstigen Preis besticht. In der Grundversion (mit zwei Laufwerken) kostet das System nur DM 1.598,–, das voll ausgebaute System mit vier Laufwerken wird DM 2.998,– kosten. (SR)

Mehr Speicherplatz

Worauf viele Schneider Anwender bereits seit langem gewartet haben, Data Media bietet es jetzt an: Auf der Internationalen Funkausstellung in Berlin wird die neue Speichererweiterung vorgestellt. Mit wenigen Handgriffen können Sie mittels einer Platine Ihren Schneider Computer zu einem professionellen Medium aufrüsten. Die Speichererweiterung wird direkt auf die CPU und das GATE ARRAY des Computers aufgesteckt und ist danach von außen nicht mehr sichtbar. Als Option, gegen einen Aufpreis, soll ein Festspeicher angeboten werden, der nach Abschalten des Computers ein im Speicher befindliches Programm über Tage bereit hält.
Die Grundversion mit 64 KB Erweiterung wird zum Preis 198,– DM angeboten, 128 KB kosten 298,– DM, 256 KB 398,– DM und für 498,– DM wird der voll aufgerüstete Speicher mit 512 KB erhältlich sein. Die kleineren Versionen sind nachträglich bis zur vollen Kapazität aufrüstbar.
Wer von vornherein nur eine Speichererweiterung um 64 KB anschließen möchte, für den gibt es eine noch preiswertere Alternative. Eine entsprechende Platine wird zum Tiefstpreis von nur 128,– DM angeboten werden. Die genannten Preise sind unverbindliche Preisempfehlungen.
Zur Auslieferung gelangen diese Erweiterungen unmittelbar nach der Internationalen Funkausstellung ab dem 10. September über den Fachhandel oder direkt bei Data Media.

LINERUNNER

Erläuterungen zum Spiel:
Spielziel ist, die gesamten Linien des Spielfeldes inklusive der beiden Plätze links und rechts des Linienfeldes durch Überlaufen mit der Spielerfigur in weiß umzufärben, wobei man stets von zwei Monstern verfolgt wird. Befindet sich die Spielerfigur auf einem der beiden angesprochene Plätze links bzw. rechts vom Spielfeld, so kann die Spielerfigur von den Monstern nicht angegriffen werden.
Ab Level 2 werden zusätzlich Hindernisse gesetzt, die von den Monstern übersprungen werden können, von der Spielerfigur jedoch nicht, es sei denn, der Spieler hatte zuvor die Feuertaste betätigt und so den „Hindernisbonus" gesetzt (Spielerfigur blinkt währenddessen, der Hindernisbonus wird z.B. dann zwingend nötig, wenn der Zugang zu einem oder mehreren noch nicht weiß eingefärbten Linienelementen durch Hindernisse versperrt ist).
Ab Level 4 sind die Monster in der Lage, ein schon weiß eingefärbtes Linienelement wieder „zurückzufärben".
Das Spiel besitzt von der Programmstruktur her keine Begrenzung der Levelanzahl; diese wird also nur durch die Fähigkeiten des Spielers beeinflußt.
Anzahl der Spielerfiguren: 3
Anzahl Hindernisbonus: 3
Angezeigt werden in Zeile 1 unter dem Spielfeld der aktuelle Punktestand und ganz rechts, dargestellt durch einen Pfeil, die verbleibende Anzahl des Hindernisbonus; dann in Zeile 2 den aktuellen Level und die verbleibenden Spielerfiguren (außer der auf dem Spielfeld).
Punktewertung:
Für jedes weiß eingefärbte Linienelement: 1 Punkt.
Bis Level 8 einschließlich für jedes fertige Bild: 100 Punkte.
Ab Level 9: je 200 Punkte.
Für jedes gesetzte Hindernis: 10 Punkte Abzug.
Für jedes vom Monster zurückgefärbte Linienelement: 5 Punkte Abzug.
Für jeden benutzten Hindernisbonus: 50 Punkte Abzug.
Anmerkung:
Das Spiel ist für Joystick-Benutzung konzipiert.
Soll das Spiel über die Tastatur gesteuert werden, so können die gewünschten Tastencodes (aus Handbuch, Anhang III/ Seite 16) ausgewählt und in die Zeilen 1300 - 1340 eingesetzt werden. Z.B. X-Richtung vorwärts soll über Taste „H" erfolgen (Tastencode 44), so muß Zeile 1310 wie folgt geändert werden:

IF INKEY (44) = 0 THEN...

Nachfolgend die Wirkung der Zeilen 1300 - 1340:
1300 "Feuerknopf": Hindernisbonus setzen
1310 Bewegung X-Richtung vorwärts
1320 Bewegung X-Richtung rückwärts
1330 Bewegung Y-Richtung hoch
1340 Bewegung Y-Richtung runter

```
10 DEFINT b-z:ENV 1,1,15,10,15,-1,40:ENV
 2,11,-1,2:ENV 3,15,-1,50:ENT 1,20,-5,2:
ENT 2,20,5,2:ENT 3,1,0,100,1,-49,100,1,-
31,100:hp=0:RANDOMIZE TIME
20 REM **** VORSPANN ****
30 MODE 0:BORDER 4:INK 0,4:PAPER 0:PEN 1
:CLS:TAG:RESTORE 280
40 EVERY 20,1 GOSUB 290:EVERY 37,2 GOSUB
 270
50 a$="LN-UNR":b$="IERNE "
60 FOR l=0 TO 100 STEP 2
70 FOR k=60 TO 560 STEP 100
80 v=(k+55)/100:ORIGIN k,180+l:PRINT MID
$(a$,v,1);
90 NEXT k
100 FOR k=110 TO 610 STEP 100
110 v=(k+5)/100:ORIGIN k,380-l+3:DRAW 25
,0,0:ORIGIN k,380-l:DRAW 0,0,1:PRINT MID
$(b$,v,1);
120 NEXT k,l
130 ff=REMAIN (1):ff=REMAIN (2):INK 1,6,
24
140 SOUND 1,770,500,15,1:SOUND 4,780,500
,15,1
150 ORIGIN 45,220:DRAWR 560,0:DRAWR 0,10
0:DRAWR -560,0:DRAWR 0,-100
160 TAGOFF:INK 2,19:PEN 2:FOR z=1 TO 100
00:NEXT z
170 LOCATE 3,2:PRINT CHR$(164);" 1985  G
.S.":FOR z=1 TO 2000:NEXT z
180 LOCATE 3,15:PRINT"Ein Spiel von":FOR
 z=1 TO 2500:NEXT z
190 LOCATE 10,21:PRINT CHR$(224):FOR z=1
 TO 500:NEXT z
200 FOR k=10 TO 19
210 LOCATE k,21:PRINT" "+CHR$(224):SOUND
 4,60,5,14
220 FOR z=1 TO 200:NEXT z,k
230 FOR k=0 TO 19
240 LOCATE 20-k,21:PRINT CHR$(224);LEFT$
("-Gerd Schroeder      ",k):SOUND 1,300,1
0,14
250 FOR z=1 TO 400-k*10:NEXT z,k
260 LOCATE 1,21:PRINT"  ":FOR z=1 TO 100
00:NEXT z:GOTO 310
270 READ n:READ m:SOUND 1,n,75,15:SOUND
4,m,75,15:RETURN
280 DATA 716,179,956,239,851,213,851,213
,358,89,478,119,426,106,426,106,478,63,4
26,71,379,80,358,89,319,95,358,89,379,80
,500,510,600,610,800,810
290 farbe=RND*26:IF farbe=4 THEN 290
300 INK 1,farbe:RETURN
310 REM **** VARIABLENBESETZUNG ****
320 DIM m(21,23),mx(2),my(2),b(2),pe(21,
23)
330 level=1:leben=3:p=0:pkte=0:fhm=3
340 REM **** FARBEN ****
350 v=INT(RND*6)+1:w=INT(RND*4)+1
360 ON v GOSUB 370,380,390,400,410,420:G
OTO 430
370 INK 1,14:RETURN
380 INK 1,7:RETURN
390 INK 1,15:RETURN
400 INK 1,16:RETURN
410 INK 1,6:RETURN
420 INK 1,2:RETURN
430 ON w GOSUB 440,450,460,470:GOTO 480
440 INK 0,0:RETURN
450 INK 0,3:RETURN
460 INK 0,1:RETURN
470 INK 0,4:RETURN
480 INK 2,26:INK 3,24:INK 4,9:INK 5,24,6
:INK 6,0:INK 8,26:INK 9,26
490 REM **** MONSTER-AUSWAHL ****
500 v=INT(RND*6)+1
510 ON v GOSUB 520,530,540,550,560,570:G
OTO 580
520 SYMBOL 255,0,129,90,60,90,102,126,10
2:RETURN
530 SYMBOL 255,126,90,90,126,24,24,36,66
:RETURN
540 SYMBOL 255,129,66,60,52,44,60,66,129
:RETURN
550 SYMBOL 255,36,90,153,24,36,66,66,0:R
ETURN
560 SYMBOL 255,68,40,16,40,40,16,40,68:R
ETURN
570 SYMBOL 255,66,165,66,189,66,165,66,0
:RETURN
580 REM **** BILDAUFBAU ****
590 RESTORE 660:CLS:PEN 1:v=RND*26:BORDE
R v
600 SOUND 1,1000,300,13:SOUND 4,1007,300
,13
610 FOR k=2 TO 22
620 FOR l=2 TO 19
630 READ m:IF m<147 THEN l=l+m:GOTO 650
640 m(l,k)=m:LOCATE l,k:PRINT CHR$(m);
```

```
650 NEXT l,k:GOTO 870
660 DATA 150,154,154,154,154,158,154,154
,154,154,158,154,154,154,154,154,154,156
670 DATA 149,3,149,3,149,5,149
680 DATA 149,3,149,3,149,5,149
690 DATA 151,154,154,154,154,155,158,154
,154,154,157,5,149
700 DATA 149,4,149,2,149,5,149
710 DATA 149,4,149,2,151,154,154,154,154
,154,154,157
720 DATA 149,4,149,2,149,5,149
730 DATA 149,4,151,154,154,154,157,5,149
740 DATA 151,154,154,154,154,154,157,2,1
51,154,154,154,154,154,154,157
750 DATA 157,4,149,2,149,5,151
760 DATA 149,4,149,2,149,5,149
770 DATA 149,4,149,2,149,5,149
780 DATA 151,154,154,154,158,154,155,154
,154,154,157,5,149
790 DATA 149,2,149,4,151,154,154,154,154
,154,154,157
800 DATA 149,2,149,4,149,5,149
810 DATA 149,2,149,4,149,5,149
820 DATA 151,154,154,154,155,154,158,154
,154,154,155,158,154,154,154,154,154,157
830 DATA 149,4,149,3,149,4,149
840 DATA 149,4,149,3,149,4,149
850 DATA 149,4,149,3,149,4,149
860 DATA 147,154,154,154,154,154,155,154
,154,154,154,155,154,154,154,154,154,153
870 LOCATE 1,11:PRINT CHR$(127):LOCATE 2
0,11:PRINT CHR$(127):m(1,11)=127:m(20,11
)=127
880 WINDOW #1,1,20,24,25:PAPER #1,4:PEN
#1,2:CLS #1
890 PRINT #1,"PUNKTE"TAB(18);STRING$(fhm
,CHR$(229)):LOCATE #1,1,2:PRINT #1,"LEVE
L";level;"LEBEN ";STRING$(leben-1,CHR$(2
24))
900 REM **** VARIABLENBESETZUNG ****
910 xx=1:yy=11:b(1)=2:b(2)=4:psoll=169:p
kte=pkte+p:p=0:FOR k=1 TO 21:FOR l=1 TO
23:pe(k,l)=1:NEXT l,k
920 FOR l=1 TO 2
930 mx(l)=INT(RND*17+2):my(l)=INT(RND*20
+2):IF m(mx(l),my(l))=0 THEN 930
940 NEXT l
950 REM **** FIGUREN SETZEN ****
960 LOCATE xx,yy:PEN 2:PRINT CHR$(224)
970 PEN 3:FOR k=1 TO 2:LOCATE mx(k),my(k
):PRINT CHR$(255):NEXT k
980 REM **** SCHWIERIGKEITSGRAD ****
990 IF level>1 AND level<6 THEN EVERY 10
00-level*100 GOSUB 1580 ELSE IF level>5
AND level<12 THEN EVERY 750-level*50 GOS
UB 1580 ELSE IF level>11 THEN EVERY 1500
 GOSUB 1580
1000 IF level>3 AND level<15 THEN EVERY
1400-level*70,1 GOSUB 1630 ELSE IF level
>14 THEN EVERY 400,1 GOSUB 1630
1010 REM **** HAUPT-ROUTINE ****
1020 DI:FOR k=1 TO 2
1030 GOSUB 1220:LOCATE #1,8,1:PRINT #1,p
+pkte:IF p=psoll THEN GOSUB 1650:GOTO 34
0
1040 LOCATE mx(k),my(k):PEN pe(mx(k),my(
k)):PRINT CHR$(m(mx(k),my(k)))
1050 IF b(k)<2 THEN GOSUB 1100 ELSE GOSU
B 1160
1060 LOCATE mx(k),my(k):PEN 3:PRINT CHR$
(255)
1070 IF (xx=mx(1) AND yy=my(1)) OR (xx=m
x(2) AND yy=my(2)) THEN 1350
1080 NEXT k:EI:GOTO 1020
1090 REM **** BEWEGUNG MONSTER HORIZONTA
L ****
1100 IF k=1 THEN IF mx(1)<xx THEN b(1)=1
 ELSE b(1)=-1
```

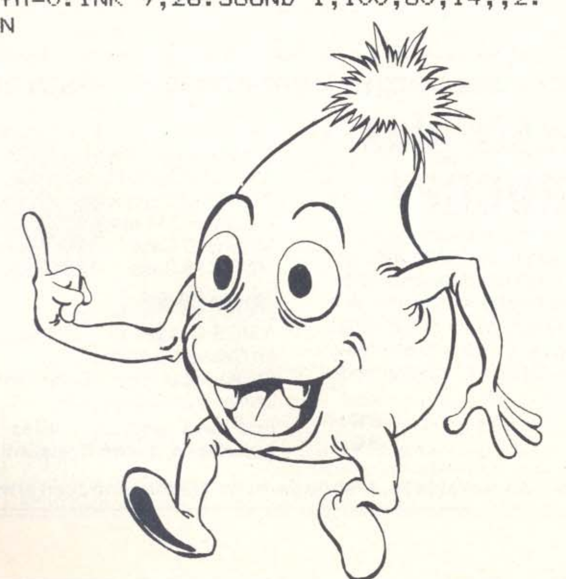




tm 4373

```
1110 IF m(mx(k),my(k)+1)>127 AND my(k)<y
y THEN my(k)=my(k)+1:b(k)=4:RETURN
1120 IF m(mx(k),my(k)-1)>127 AND my(k)>y
y THEN my(k)=my(k)-1:b(k)=2:RETURN
1130 IF m(mx(k)+b(k),my(k))>127 THEN mx(
k)=mx(k)+b(k):RETURN
1140 GOSUB 1160
1150 REM **** BEWEGUNG MONSTER VERTIKAL
****
1160 IF k=2 AND xx<>1 AND xx<>20 THEN IF
 my(2)<yy THEN b(2)=4 ELSE b(2)=2
1170 IF m(mx(k)+1,my(k))>127 AND mx(k)<x
x THEN mx(k)=mx(k)+1:b(k)=1:RETURN
1180 IF m(mx(k)-1,my(k))>127 AND mx(k)>x
x THEN mx(k)=mx(k)-1:b(k)=-1:RETURN
1190 IF m(mx(k),my(k)+b(k)-3)>127 THEN m
y(k)=my(k)+b(k)-3:RETURN
1200 GOSUB 1100
1210 REM **** BEWEGUNG SPIELERFIGUR ****
1220 xv=0:yv=0
1230 IF INKEY(76)=0 THEN GOSUB 1920
1240 IF INKEY(75)=0 THEN xv=1:GOTO 1290
1250 IF INKEY(74)=0 THEN xv=-1:GOTO 1290
1260 IF INKEY(73)=0 THEN yv=1:GOTO 1290
1270 IF INKEY(72)=0 THEN yv=-1:GOTO 1290
1280 RETURN
1290 IF m(xx+xv,yy+yv)=0 OR (m(xx+xv,yy+
yv)=202 AND fh=0) THEN RETURN
1300 LOCATE xx,yy:IF fh AND pe(xx,yy)=5
THEN PEN 5 ELSE PEN 8:pe(xx,yy)=8
1310 PRINT CHR$(m(xx,yy))
1320 xx=xx+xv:yy=yy+yv:LOCATE xx,yy:PEN
9:PRINT CHR$(224)
1330 IF pe(xx,yy)<>1 THEN SOUND 2,200,2,
13 ELSE p=p+1:SOUND 2,70,2,13
1340 IF (xx=mx(1) AND yy=my(1)) OR (xx=m
x(2) AND yy=my(2)) THEN 1350 ELSE RETURN
1350 REM **** SUBROUTINE KOLLISION ****
1360 ff=REMAIN(0):ff=REMAIN(1):ff=REMAIN
(1):fh=0:SOUND 2,800,400,,3,,10
1370 PEN 5:LOCATE xx,yy:PRINT CHR$(166):
FOR z=1 TO 1000:NEXT z
1380 FOR l=yy TO 22
1390 LOCATE xx,yy:PRINT " "
1400 yy=yy+1:LOCATE xx,yy:PRINT CHR$(225
):FOR z=1 TO 200:NEXT z,l
1410 BORDER 0:PEN 2
1420 FOR l=22 TO 2 STEP -1
1430 LOCATE 10,l:PRINT CHR$(143):NEXT l
1440 LOCATE 6,8:PRINT STRING$(9,CHR$(143
)):LOCATE 10,8:PEN 3:PRINT CHR$(225)
1450 RESTORE 1500
1460 FOR l=1 TO 21
1470 READ u,v,w:SOUND 1,u,w,15:SOUND 4,v
,w,15
1480 NEXT l
1490 leben=leben-1:FOR z=1 TO 4000:NEXT
z:GOTO 1520
1500 DATA 638,80,80,0,0,20,638,80,70,0,0
,20,638,80,20,0,0,5,638,80,80,0,0,30
1510 DATA 536,67,60,0,0,15,568,71,30,0,0
,5,568,71,50,0,0,15,638,80,30,0,0,5,638,
80,50,0,0,15,716,89,30,0,0,5,638,80,100
1520 REM **** BILDSCHIRM LOESCHEN ****
1530 PEN 6:FOR ze=1 TO 25
1540 FOR sp=1 TO 20
1550 LOCATE sp,ze:PRINT CHR$(143)
1560 NEXT sp:SOUND 2,ze*20,5,13:NEXT ze
1570 IF leben=0 THEN 1750 ELSE 340
1580 REM **** HINDERNISSE SETZEN ****
1590 DI:xh=RND*17+2:yh=RND*20+2:IF m(xh,
yh)=0 OR m(xh,yh)=202 THEN 1590
1600 pkte=pkte-10:IF pe(xh,yh)<>8 THEN p
soll=psoll-1
1610 m(xh,yh)=202:LOCATE xh,yh:PEN 5:pe(
xh,yh)=5:PRINT CHR$(202):SOUND 4,50,20,1
4:RETURN
1620 REM **** MONSTER SETZT ALTE FARBE *
```

```
1630 IF pe(mx(1),my(1))=8 THEN pe(mx(1),
my(1))=1:pkte=pkte-5:psoll=psoll+1:SOUND
 4,400,30,14:RETURN ELSE RETURN
1640 REM **** NAECHSTER LEVEL ****
1650 ff=REMAIN(0):ff=REMAIN(1):ff=REMAIN
(2):fh=0:INK 8,0,26:level=level+1:IF lev
el<9 THEN pkte=pkte+100 ELSE pkte=pkte+2
00
1660 FOR k=1 TO 5:FOR l=160 TO 30 STEP -
2
1670 SOUND 2,1,1,15:INK 0,k:BORDER k
1680 NEXT l,k
1690 FOR k=1 TO 26:FOR l=160 TO 60 STEP
-10
1700 SOUND 2,1,1,15:INK 0,k:BORDER k
1710 NEXT l,k
1720 INK 0,0:INK 1,6,26:BORDER 0:PEN 1:C
LS
1730 LOCATE 5,10:PRINT"L E V E L":LOCATE
 8,15:PRINT level
1740 SOUND 1,239,300,14,,3:FOR z=1 TO 60
00:NEXT z:RETURN
1750 REM **** SPIELENDE ****
1760 SPEED INK 30,30:BORDER 15,6:INK 0,6
,15:INK 2,0:CLS
1770 PEN 2:LOCATE 2,1:PRINT"S P I E L E
N D E"
1780 LOCATE 1,5:PRINT"HOECHSTPUNKTZAHL:"
:LOCATE 7,7:PRINT hp
1790 LOCATE 1,11:PRINT"IHRE PUNKTZAHL:":
LOCATE 7,13:PRINT p+pkte
1800 IF (p+pkte)<hp THEN FOR k=60 TO 400
 STEP 5:SOUND 1,k,5,14:NEXT k:GOTO 1860
1810 LOCATE 1,15:PRINT"GLUECKWUNSCH !!":
hp=p+pkte
1820 RESTORE 1910
1830 FOR k=1 TO 16
1840 READ v:SOUND 1,v,50,14
1850 NEXT k
1860 PEN 3:LOCATE 1,22:PRINT"NEUES SPIEL
 ?"
1870 LOCATE 1,24:PRINT"TASTE <J/N> DRUEC
KEN"
1880 a$=INKEY$:IF a$="j" OR a$="J" THEN
1900
1890 IF a$<>"n" AND a$<>"N" THEN 1880 EL
SE END
1900 SPEED INK 10,10:GOTO 330
1910 DATA 179,239,213,0,89,119,106,0,200
,200,250,250,300,300,280,280
1920 REM **** HINDERNIS-BONUS ****
1930 IF level<2 OR fhm=0 OR fh THEN RETU
RN ELSE fh=1:pkte=pkte-50:INK 9,6,18:SOU
ND 1,200,60,14,,1:AFTER 500,2 GOSUB 1940
:fhm=fhm-1:LOCATE #1,18,1:PRINT #1,STRIN
G$(fhm,CHR$(229));STRING$(ABS(fhm-3)," "
):RETURN
1940 REM **** ENDE HINDERNIS-BONUS ****
1950 fh=0:INK 9,26:SOUND 1,100,60,14,,2:
RETURN
```



Ein Typenrad- und zwei Matrixdrucker standen auf dem Prüfstand in der CPC-Redaktion. Merkmale und Leistungsdaten jedes einzelnen finden Sie in dem folgenden Testbericht.



Drucker im Test: Die Matrixdrucker Quen Data DMP 1100, DMP 1800 und der Typenrad-Drucker Brother HR-1

Brother HR-10

Der Typenraddrucker HR-10 wurde erstmals auf der diesjährigen Hannover Messe vorgestellt. Schon vom Äußeren unterscheidet sich der HR-10 von seinen Kollegen: Er ist wesentlich stilvoller gebaut, wiegt nur etwa 6 kg und ist auch in seinen Abmessungen [410(B)x288(H)x131(D)] kleiner als übrige Typenrad-Drucker. Der an der unteren Vorderseite angebrachte Tragegriff und die Vorrichtung zur Aufnahme des Stromkabels lassen einen problemlosen Transport zu. Der HR-10 erreicht eine Druckgeschwindigkeit von 12 Zeichen pro Sekunde (bidirektional). Das Druckgeräusch ist wie bei allen Typenrad-Druckern auf Dauer ziemlich nervend. Zieht man jedoch das gestochen scharfe Druckbild in Betracht, wird schnell klar, daß der HR-10 ein Korrespondenz-Drucker ist und sein will. Durch die austauschbare Typenscheibe stehen dem Anwender eine Reihe von verschiedenen Schriftarten zur Verfügung, das Wechseln von Typenrad und Farbband ist kinderleicht vorzunehmen. Neben der bei Typenrad-Druckern serienmäßigen Einzelblattverarbeitung kann der HR-10, mit entsprechendem Aufsatz, auch Endlospapier verarbeiten. Diese Traktorführung ist als Option erhältlich. Für Textverarbeitung stellt der HR-10 einige Funktionen zur Verfügung, diese entnehmen Sie bitte unserem Testausdruck. Vorzüglich ist die Anordnung der DIP-Schalter, die neben dem Interface-Stecker an der Rückseite des Druckers leicht zugänglich sind. Vorbildlich ist auch das Bedienungshandbuch, mit 117 Seiten und klarer Gliederung läßt es keinen Wunsch offen. Zum Preis von ca. DM 1.150,- erhält man einen professionellen Drucker, der sicher überall dort seine Verwendung finden wird, wo ein exzellentes Schriftbild bei geringem Kostenaufwand erreicht werden soll.

Quen Data DMP-1100:

Der DMP 1100 reiht sich nahtlos in die Reihe der leistungsfähigen Matrix-Drucker ein. Mit einer Druckgeschwindigkeit von 100 Zeichen/Sekunde liegt er im Mittelmaß, besticht aber durch sein für Matrixdrucker scharfes Druckbild. Dadurch wird er auch für semipro-

DAS SCHRIFTBILD DES BROTHER HR 10
IST WIE BEI ALLEN TYPENRAD-DRUCKERN SEHR SAUBER
MAN KANN UNTERSTREICHEN
IM DOPPELDRUCK ARBEITEN
ODER MIT SCHATTENDRUCK HERVORHEBEN



r alle Fälle

0 (v.l.n.r.)

fessionelle Anwendungen interessant, zumal das Druckgeräusch sehr leise ist. Desweiteren vereinigt der DMP 1100 eine Menge Features der Zeichendarstellung in sich. Mit einfachen Steuerbefehlen lassen sich nicht nur die Schriftarten 'Pica' und 'Elite' anwählen, der DMP 1100 stellt noch folgende Darstellungsarten zur Verfügung:

- Zeichenvergrößerung, Zeichenverdichtung, Zeichenhervorhebung, Doppelanschlag, Hoch- und Tiefstellung, Unterstreichen, 8 internationale Zeichensätze, ladbarer Zeichengenerator, Bit-Image Mode zur Grafikdarstellung.

Eine Besonderheit ist die Papierverarbeitung, denn sowohl Endlospapier als auch Einzelblätter können verarbeitet werden. Der DMP 1100 hat einen eingebauten RAM-Puffer, der optional auf 2KB bzw. 4KB erweitert werden kann.

Erwähnenswert ist auch die Möglichkeit des Hex Dump-Druckes, bei dem die Zeichen in hexagonaler 8bit-Weise ausgedruckt werden. Umständlich dagegen erschien uns die Plazierung der DIP-Schalter, die ohne ein Öffnen des Druckergehäuses nicht zugänglich sind. Das Kassetten-Farbband ist lediglich zum einmaligen Gebrauch gedacht, doch können laut Hersteller bis zu 8 Millionen Zeichen pro Band gedruckt werden. Im Test erwies sich der DMP 1100 als sehr zuverlässig – auch mit der eigentümlichen Schneider-Grafik gab es keine Probleme beim Ausdruck. Für ca. DM 898,- erhält man einen leistungsfähigen Matrixdrucker zu einem wirklich günstigen Preis, der einige professionelle Features bietet.

Das ca. 100 Seiten umfassende, deutsche Handbuch läßt den Umgang mit diesem Drucker problemlos vonstatten gehen.

```
DER DMP-1100 VON QUEN DATA
DRUCKT SCHNELL
UND SAUBER
ER KANN HERVORHEBEN
SOWIE IM COMPRESSED MODE DRUCKEN
AUSSERDEM UNTERSTREICHEN
ES GIBT DIE SCHRIFTARTEN 'ELITE'
 UND 'PICA'
```

DMP 1800 LQ

Der zweite Quen Data Drucker auf dem CPC-Prüfstand war der DMP 1800 LQ. Das LQ hinter der Bezeichnung deutet schon auf die Near-Letter-Quality des Druckbildes hin – allerdings übertraf der erste Testausdruck unsere kühnsten Erwartungen. Der DMP 1800 ist mit einer Druckgeschwindigkeit von 180 Zeichen/Sekunde der absolut schnellste Matrixdrucker, der bisher in unserer Redaktion vorlag. Dabei ist das Schriftbild, auch ohne NLQ-Modus, gestochen scharf und genügt damit auch professionellen Ansprüchen. Wie auch beim DMP 1100, ist die wahlweise Verarbeitung von Endlos- und Einzelblattpapier möglich – der Papiereinzug ist problemlos und leicht vorzunehmen. Der DMP 1800 arbeitet mit einem Textil-Farbband, das wesentlich länger hält und wieder aufgefrischt werden kann.

```
AUCH DER DMP-1800 VON QUEN DATA
DRUCKT SCHNELL
UND MIT FETTSCHRIFT
ER KANN HERVORHEBEN
SOWIE IM COMPRESSED MODE DRUCKEN
AUSSERDEM UNTERSTREICHEN
ES GIBT DIE SCHRIFTARTEN 'ELITE'
 UND 'PICA'
```

Der Zeichenvorrat umfaßt neben sämtlichen, schon beim DMP-1100 erwähnten Features auch die Kursivschrift, die Steuersequenzen sind bei beiden Druckern identisch.

Vorbildlich ist die Anordnung der DIP-Schalter, neben der Schnittstellenbuchse sind diese leicht zugänglich und umzuschalten. An den bei Matrixdrukkern üblichen LED-Anzeigen der Gehäuseoberseite läßt sich jederzeit der Betriebszustand des Druckers abfragen, wobei der DMP 1800 eine automatische Papierende-Erkennung besitzt. In der mehrwöchigen Testphase erwies sich dieser Drucker als sehr zuverlässig, besticht durch einfache Bedienung und last but not least, durch seine enorme Geschwindigkeit. Mit dem eingebauten 3K-Datenpuffer und der Geschwindigkeit von 180Z-S wird der Computer nicht unnötig blockiert und ein schnelles, effektives und professionelles Arbeiten wird möglich.

Der Porsche unter den Matrixdruckern wird zum Preis eines 'Käfers' angeboten. Mit ca. DM 1498,– liegt er deutlich unter dem Preis vergleichbarer Drucker. (SR)

Basic-Kurs Teil 7

Herzlich Willkommen zum siebten Teil unseres Basic-Kurses.
In der letzten Lektion hatten wir sämtliche Programmierkenntnisse auf einen Haufen gelegt, und mit den wenigen uns schon bekannten Befehlen das erste Spielprogramm geschrieben.

Das nächste Projekt, das wir angehen wollen, wird ein Spiel mit Bewegung sein. Die Programmierung dieses Spiels wird sich über mehrere Lektionen erstrecken. Das endgültige Spiel wird eine Basic-Version des bekannten Spielhallenspiels BREAKOUT, in dem ein Ball, der sich auf dem Bildschirm bewegt, mit einem Schläger, welcher vom Spieler gesteuert wird, abgeschlagen und zerstört dabei eine Mauer, die sich am oberern Bildschirmrand befindet. In dieser Lektion werden wir nun einen neuen Befehl erlernen, der es uns in Verbindung mit ein paar IF-THEN-Anweisungen sowie einigen PRINT-Befehlen ermöglicht, eine Bewegung auf dem Bildschirm darzustellen.
Der neue Befehl, den wir dazu kennenlernen, heißt LOCATE.
LOCATE bedeutet in der deutschen Übersetzung: Lokalisiere und positioniere den Cursor auf einer bestimmbaren Bildschirmstelle. Eingegeben wird er in der Form:

LOCATE X-KOORDINATE,Y-KOORDINATE

Wird nach einer Locate-Anweisung ein PRINT-Befehl ausgesprochen, so erscheint das zu printende Zeichen an der vorher bestimmten Position.
Bei einer LOCATE-Anweisung können Variablen übergeben werden.
Zur Programmierung unseres Spiels definieren wir zunächst einmal ein Spielfeld. Dazu löschen wir in der ersten Zeile den Bildschirm und schreiben:

10 CLS

Danach lassen wir die obere Umrandung printen, die auch gleichzeitig die Mauer enthält, die zerstört werden muß.
Als Umrandung verwenden wir das Zeichen '!' und als Mauersteine das Doppelgatter '#'. Das Zeichen 'I' bekommen wir, wenn wir gleichzeitig die Shift-Taste und den Klammeraffen drücken.

Zeile 20 lautet also:

20 PRINT "I # # # # # # # # # # I"

Wir haben in dieser Zeile ein Umrandungs-Symbol, gefolgt von 10 Mauersteinen und einem abschließenden Umrandungs-Symbol.
Um die Umrandung nach unten fortzuführen, printen wir 11 weitere Zeilen, die die Umrandungs-Symbole enthalten und anstelle der Mauersteine Leerzeichen.
Die Zeilen 30 bis 130 sehen also folgendermaßen aus:

30 PRINT "I   I"
.
130 PRINT "I   I"

Nach Ablauf mit RUN sehen Sie, wie der Bildschirm aufgebaut wird und ein 12*12 Zeichen großes Feld belegt.
Nun wollen wir in der oberen linken Ecke des Spielfeldes den Ballstartpunkt festlegen. Die Positionierung des Balls basiert auf 4 Variablen mit den Namen BALLX,BALLY,MOVEX, MOVEY.
BALLX und BALLY enthalten die momentane Position des Balls,
MOVEX und MOVEY enthalten die momentane Bewegungsrichtung des Balls. Wir setzen den Ball also auf die Startposition und belegen BALLX und BALLY vor. Die Eck-Koordinate ist 2,2 und wir schreiben:

140 BALLX=2:BALLY=2

Die Bewegungen des Balls sollen positiv verlaufen und in 1er Schritten ablaufen.

150 MOVEX=1:MOVEY=1

Danach wird die Stelle des Bildschirms lokalisiert und zunächst einmal ein Leerzeichen, welches als Löscher für den Ball benutzt wird. Die nächsten beiden Zeilen sind also:

160 LOCATE BALLX,BALLY
170 PRINT " "

Nun muß die Bewegungsrichtung zur Ballposition addiert werden. Je nachdem, ob die Bewegungsvariable positiven oder negativen Charakter hat, wird der Ball nach oben oder unten bzw. rechts oder links bewegt. Danach locaten wir die neue Ballposition und printen den Ball.

180 BALLX=BALLX+MOVEX:
BALLY=BALLY+MOVEY
190 LOCATE BALLX,BALLY
200 PRINT "O"

Wenn wir das Programm nun starten, bewegt sich der Ball auf Position 3,3. Wollen wir nach 160 springen, erhöht sich die Position wiederum um 1. Allerdings schießt der Ball dann irgendwann über das Spielfeld hinaus, da ihm die Randbegrenzung fehlt. Diese Begrenzung muß festlegt und mit einer Bewegungsänderung koordiniert werden. Die Grenzen unseres Spielfeldes in X-Richtung würden logischerweise bei 1 und 12 liegen. Geben wir diese Werte jedoch als Grenze ein, so wird jedesmal, wenn der Ball diese erreicht, ein Zeichen unserer Umrandung gelöscht. Wir begrenzen den Bildschirm also auf 2 und 11. Erreicht der Ball einen dieser Ränder, so muß ihm eine Änderung der Bewegung mitgeteilt werden. Bewegen wir den Ball mit einer positiven X-Richtung, so nähert er sich dem 11er Rand. Hat er diesen erreicht, muß seine Bewegung negativ werden. In die andere Richtung funktioniert es genauso. Auch hier müssen beim Auftreffen auf die Begrenzung, die Vorzeichen der Richtungsvariable geändert werden. Wir schreiben als zwei IF-THEN-Anweisungen.

210 IF BALLX=2 THEN MOVEX=1
220 IF BALLX=11 THEN MOVEX=-1

Genauso verhält es sich bei der Y-Positionierung. Hierbei liegt die Obergrenze jedoch bei 12 und 1, da die Mauer mit den Steinen entfernt werden soll. Die beiden nächsten Zeilen lauten also:

230 IF BALLY=1 THEN MOVEY=1
240 IF BALLY=12 THEN MOVEY=-1

Damit ist unsere Bewegung am Bildschirm schon abgeschlossen. Um die Funktion zu überprüfen, geben Sie Zeile 241 wie folgt ein:

241 GOTO 160

Starten Sie nun, werden Sie feststellen, daß sich der Ball auf dem Bildschirm hin und her bewegt. Mehr passiert allerdings noch nicht, da die Steuerung des Schlägers erst im nächsten Teil erläutert wird. Bis dahin, viel Spaß!

```
10 MODE 1
20 PRINT"|##########|"
30 PRINT"|          |"
40 PRINT"|          |"
50 PRINT"|          |"
60 PRINT"|          |"
70 PRINT"|          |"
80 PRINT"|          |"
90 PRINT"|          |"
100 PRINT"|          |"
110 PRINT"|          |"
120 PRINT"|          |"
130 PRINT"|          |"
140 ballx=2:bally=2
150 movex=1:movey=1
160 LOCATE ballx,bally
170 PRINT" "
180 ballx=ballx+movex:
bally=bally+movey
190 LOCATE ballx,bally
200 PRINT"O"
210 IF ballx=2 THEN movex=1
220 IF ballx=11 THEN movex=-1
230 IF bally=1 THEN movey=1
240 IF bally=12 THEN movey=-1
241 GOTO 160
```

(TM)

Nachdem wir ausführlich auf die Computer-Entwicklungen der zurückliegenden Jahre eingegangen sind, wenden wir uns zunächst dem großen Metier Software zu. Im besonderen werden wir auf die verschiedenen Programmiersprachen eingehen und im geschichtlichen Überblick deren Entwicklungen und Anwendungen aufzeigen.

Die Geschichte der Mikroelektronik

Teil III

Foto IBM

Mit dem von J. v Neumann erfundenen Konzept der Speicherprogrammierung wurde auch das Thema Programmiersprachen interessant. Zwischen 1950 und 1970 war dann auch ein regelrechter BOOM zu verzeichnen – immer neue Programmiersprachen wurden entwickelt. Pfiffige Programmierer erlebten zu dieser Zeit ihre absoluten Höhepunkte und als Ergebnis waren ausgereifte Computersprachen zu vermelden, die auch heute noch in aller Welt ihre Verwendung finden.
Allerdings war der erste, uns bekannte Programmierer eine Frau und deren Tätigkeit liegt bereits um knapp 150 Jahre zurück. Der erste Programmierer der Welt war Lady Augusta Ada Lovelace (1816-1852), die Tochter des Dichters Lord Byron. Mit ihrer Arbeit unterstützte sie vorrangig Charles Babbage, der zu dieser Zeit gerade mit der Entwicklung der Differenziermaschine und der Analytischen Maschine beschäftigt war. Ada war Babbage behilflich, einige Fehler und Irrtümer in seiner Arbeit zu finden und hatte zu dieser Zeit bereits einige zukunftsweisende Ideen, die erst Jahre später Verwendung fanden. So entwickelte sie den Begriff des für eine Berechnung ständig wiederholten Befehls – heute als Schleife bzw. Subroutine bekannt. Eine im Auftrag des amerikanischen Verteidigungsministeriums in den 70'er Jahren entwickelte Programmiersprache erhielt den Namen ADA, zu Ehren der ersten Programmiererin der Welt.
Erst hundert Jahre später, als die Computer bereits mit Strom, Röhren und Relais arbeiteten, wurden die nächsten Schritte eingeleitet. Und wieder war es eine Frau, die Schlagzeilen machte. Grace Murray Hopper programmierte 1943 den Mark I, der aufgrund Babbages Entwicklungen des vorangegangenen Jahrhunderts endlich im Jahre 1944 fertiggestellt wurde. Sie schrieb die kompliziertesten Befehle, die schließlich den Mark I zum Funktionieren brachten und somit das ganze Projekt vor der Einstampfung rettete. Ihre Grundüberlegung ging von einem einzigen Programm aus, das einen grundlegenden Teil einer immer wiederkehrenden Arbeit erledigte. Daraus entstand zunächst die Programmiersprache Flow-Matic, aus deren Grundlagen anschließend die noch heute bekannte und oft verwendete COBOL-Sprache wurde.

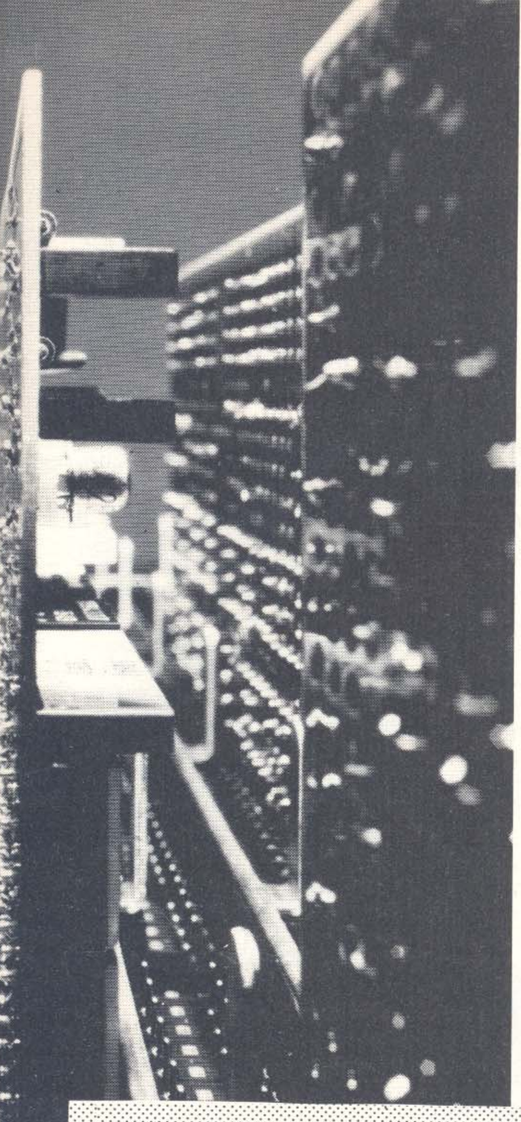
Beispiel für den Zahlenwert 671 in binärer Schreibweise	
$1 \cdot 2^9$	(512)
$0 \cdot 2^8$	(0)
$1 \cdot 2^7$	(128)
$0 \cdot 2^6$	(0)
$0 \cdot 2^5$	(0)
$1 \cdot 2^4$	(16)
$1 \cdot 2^3$	(8)
$1 \cdot 2^2$	(4)
$1 \cdot 2^1$	(2)
$1 \cdot 2^0$	(1)
1010011111	(671)

Umrechnungstabelle

Binär	Dezimal
2^4 2^3 2^2 2^1 2^0	10^2 10^1 10^0
16 8 4 2 1	100 10 1
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12

COBOL steht für „Commercial and Business-Oriented Language" und wurde 1959 in Teamarbeit entwickelt. Ziel war es, eine für alle Computersysteme einheitliche Programmiersprache zu entwickeln und somit ein Sprachen-„Torso" zu vermeiden.

Daneben gab es noch eine Reihe von Versuchsprojekten, die sich mit der Entwicklung von problem-orientierten Programmiersprachen befaßten. So entstand 1958 die Sprache ALGOL, die „Algorithmic Language". ALGOL wurde für technisch-wissenschaftliche Anwendungen entwickelt und gilt als sogenannte höhere Programmiersprache. Als Erfinder von ALGOL gilt



Peter Naur, der mit seinen Kollegen über mehrere Jahre hinweg an dieser Entwicklung erfolgreich arbeitete.

Die erste problem-orientierte Programmiersprache wurde aber bereits 1954 von John Backus entwickelt. Der bei dem Branchenriesen IBM beschäftigte Backus ist der Erfinder von Fortran, eine speziell für technisch-wissenschaftliche Zwecke entwickelte, höhere Programmiersprache. Seit ca. 1957 ist Fortran weltweit verbreitet und unterstützt heute – nachdem mehrere Fortran-Versionen aus dem Grundprogramm entstanden – neben wissenschaftlichen, vorrangig wirtschaftliche Anwendungen.

In diesen Jahren wurde noch eine weitere Programmiersprache auf einem IBM-Rechner entwickelt. John Mc Carthy erarbeitete von 1956 bis 1958 das Konzept der Programmiersprache LISP, dessen Installation bereits 1958 erfolgte. LISP steht für „List Processor" und wurde an dem weltberühmten Massachusetts Insitute of Technology (MIT) auf einem IBM 704 entwickelt. Es ist eine höhere Programmiersprache, die eine Verbindung zwischen Datenlisten- und Tabellen herstellen sollte und damit bereits Anzeichen von künstlicher Intelligenz zeigte und nichtnumerische Aufgabenstellungen bewältigen konnte.

Mit einem gänzlich anderen Ziel ging Jules Schwartz von der System Development Corp. an die Entwicklung von Jovial. Diese im Jahr 1960 entwickelte Programmiersprache wurde ausschließlich für die Entwicklung von Computer-Software für die Luftabwehr eingesetzt. Jovial steht für ‚Jules' Own Version of the International Algebraic Language und wird auch heute ausschließlich zu militärischen Zwecken eingesetzt.

Zum Entwerfen von Programmen war dann 1962 die Programmiersprache APL gedacht, die Kenneth Iverson an der Harvard-Universität entwickelte. ‚A Programming Language' (APL) ist ein Nebenprodukt des Unterrichtens an der o.g. Hochschule und wird heute vorrangig für mathematische Aufgabenstellungen verwendet.

Eine von IBM in Auftrag gegebene Entwicklung wurde dann 1964 präsentiert: PL/1 war eine völlig neue Programmiersprache für den IBM 360 und bedeutet ganz einfach ‚Programming Language'. Die von Bruce Rosenblatt und George Radin von Standard Oil entwickelte Sprache weist Elemente von COBOL und ALGOL auf und ist wirtschaftlich ausgerichtet.

Nur ein Jahr später wurde dann die Programmiersprache erfunden, die heute fast jeder Rechner versteht und die größte Publizität errang: BASIC.

Der ‚Beginners All-Purpose Symbolic Instruction Code' wurde von John Kemeny und Tom Kurtz an der Uni Dartmouth entwickelt, und erstmals am 1. Mai 1964 in einem Rechner installiert. BASIC zeichnet sich durch seine leichte Erlernbarkeit aus, was auch die Grundüberlegung der Entwickler war. Heute gilt BASIC als „Einsteigersprache" und wird daneben vorrangig für kaufmännische Anwendungen eingesetzt. Kritiker dieser recht eigentümlichen Programmiersprache sagen BASIC zwar „Verunglimpfungen der Computersprache nach", doch hat sich BASIC inzwischen zu einem nicht mehr wegzudenkenden, weltweiten Standard entwickelt.

Im Jahr 1971 entwickelte dann Niklaus Wirth an der eidgenössischen technischen Hochschule der Schweiz die Programmiersprache PASCAL. Sie wurde nach Blaise Pascal benannt und hauptsächlich zum systematischen Programmieren entwickelt. Pascal beinhaltet Elemente von ALGOL und PL/1 und wird für mathematisch-technisch-wissenschaftliche Zwecke eingesetzt. Inzwischen gibt es auch hier mehrere Versionen von Pascal, die sich aber alle durch hervorragende Programmstruktur auszeichnen.

Wir hoffen, Ihnen mit dieser Übersicht einen kleinen Einblick in die verschiedenen Programmiersprachen gegeben zu haben. Die Unterschiede der Programmierung würde hier zu weit führen. Auch in unserer nächsten Ausgabe werden Sie wieder einiges aus der Geschichte, der heute (fast) unentbehrlich gewordenen Computer erfahren.

(SR)

WASCHTEMP. LESEN
JA
WASCH ART=1 ?
NEIN
WASSER NIVEAU 2
WASSER NIVEAU 3
JA
WASCH TEMP 95° ?
NEIN
JA
WASCH ART=1 ?
NEIN
JA
TEMP = 60°C ?
NEIN

Graphic Siemens-Mikroelektronik

Ausschnitt aus einem Flußdiagramm für eine Waschmaschinensteuerung mit dazugehörigem Quellenprogramm.

Quellenprogramm (Assembler)		Objektprogramm in Maschinensprache (Binärzahlen)
...		
IN	2	0001000010101010 1101101100000010
MOV	TEMP, A	0001000010101100 00100111
MVI	A, 1	0001000010101101 1011111000000001
CMP	FLAG 1	0001000010101111 10111001
JZ	EQUAL	0001000010110000 110010101101100100010000
LDA	NIV 2	0001000010110011 001110101110000000010100
JMP	STOR 1	0001000010110110 110000111011110000010000
NOP		00000000
LDA	NIV 3	0001000010111001 001110101110000100010100
NOP		00000000
STA	R 54	0001000010111100 001100101110001000010100
MVI	A, 95	0001000010111111 1011111001011111
CMP	TEMP	0001000011000001 10111000
JZ	TMP 95	0001000011000010 110010101101000000010000

Bruce H. Hunter
50 Pascal-Programme
Sybex Verlag 1985
317 Seiten
ISBN 3-88745-065-5
Preis: 48,– DM



Die Anwender der verschiedenen, für jeden CPC angebotenen, Pascal-Versionen mußten auf geeignete Literatur entweder lange warten oder gänzlich verzichten.

Mit diesem Buch präsentiert sich nun eine Pascal-Programm-Sammlung, die sich sowohl an private als auch an kommerzielle Anwender richtet. Bei den 50 abgedruckten Programmen und den reichhaltigen und verständlichen Dokumentationen werden allerdings Grundkenntnisse vorausgesetzt, so daß für Anfänger dieses Buch weniger geeignet scheint.

Durch die klare Strukturierung ist eine Anpassung der Programme an eigene Bedürfnisse leicht möglich.

Nachfolgend einige Beispielprogramme aus dem Inhalt:

- Geschäftsprogramme: Bilanz
- Finanzprogramme: Darlehensberechnung
- Persönliche Programme: Telefonregister
- Hilfsprogramme: Ausdruck von Textdateien
- Elektrizitätsprogramme: Ohmsches Gesetz
- Ingenieur-Programme: Trägheitsmoment etc.
- Spiele: Mathespiel

Für all diejenigen, die mit Pascal arbeiten eine interessante Literatur, die reichlich Denkanstöße gibt. Für Pascal-Interessierte eine umfangreiche Programmsammlung mit interessanten Beispiel-Listings aus nahezu allen Anwendungsbereichen.

Boris Allan
Logo – selbst gelernt
MVG Moderne Verlagsgesellschaft 1984
130 Seiten
ISBN 3-478-09092-X
Preis: 29,80 DM

Die Programmiersprache LOGO wird oft sehr stiefmütterlich behandelt. Gilt sie doch als Programmiersprache für Kinder und kaum ein ernsthafter Programmierer beschäftigt sich mit dieser doch sehr interessanten Computer-Sprache.

Mit diesem Buch erhält man nun eine Einführung in LOGO – die für viele un-



bekannte Sprache. Aber gerade für die vielen Besitzer des Schneider Diskettenlaufwerkes ist LOGO nicht gerade uninteressant, liegt beim Kauf der DDI-1 doch eine entsprechende Version gratis bei. Aufgrund mangelnder Hinweise im Floppy-Handbuch ist der LOGO-Interessierte also auf zusätzliche Literatur angewiesen, die im großen und ganzen recht dünn gesät ist. In diesem Buch wird der Interessierte zielbewußt in LOGO eingeführt. Schon nach kurzer Zeit ist man in der Lage, einige Ideen zu verwirklichen. Zahlreiche Beispiele und leicht verständliche Dokumentation machen dieses Buch zu einer interessanten Lehr- und Lernliteratur.

Rodnay Zaks
Programmierung des Z-80
Sybex-Verlag
7. Auflage
640 Seiten
ISBN 3-88745-099-X
Preis: 58,00 DM

Dieses für Maschinensprache-Programmierer zum Standardwerk gewordene Buch erscheint nunmehr in der 7. Auflage mit wesentlichen Ergänzungen. Der inzwischen weltweit bekannte Autor R. Zaks zeigt, wie Programme in Maschinensprache geschrieben werden und gibt eine komplexe Einführung in die Arbeitsweise eines Z-80 Mikroprozessors. Anhand zahlreicher und leicht verständlicher Beispielaufgaben wird der Lernende Stück für Stück in die Geheimnisse seines Rechners eingeweiht und anschließend in der Lage sein, selbst Maschinen-Programme zu entwickeln. Neben dem kompletten Z-80 Befehlssatz findet der Anwender detaillierte Hinweise zur besseren Ausnutzung von Programmiertechniken sowie ein übersichtliches Stichwortverzeichnis. Zu erwähnen ist, daß die Lösungen zu den Beispielaufgaben im Anhang zusammengefaßt sind, eine direkte Lernkontrolle ist somit möglich. Dieses Buch wendet sich an alle Maschinensprache-Interessierten und diejenigen, die es noch werden wollen. Es ist nicht nur als Lehr- und Lernbuch bestens geeignet, sondern dient auch für Fortgeschrittene und Profis als kompetentes Nachschlagewerk.



Audrey und Owen Bishop
Praktische Programme für den Schneider CPC
MVG-Moderne Verlagsgesellschaft 1984
176 Seiten
ISBN 3-478-09230-2
Preis: 29,80 DM



Listing-Bücher für den CPC gibt es inzwischen reichlich.

Doch eines gleich vorweg: Dieses Buch hält, was es verspricht! Insgesamt findet man 14 Programme, die jeder für Haushalt, Hobby, Beruf oder Geschäft nutzen und anwenden kann. Dabei wurde auf eine praxisnahe Gestaltung und Zusammenstellung der Programme geachtet. Nachfolgend eine kurze Auswahl der angebotenen Programme:

- Tabellenkalkulation, Lagerverwaltung, Buchführung, Datenbank, Torten-Diagramm, Supersound, Toto-Tip, Raumplaner uvm.

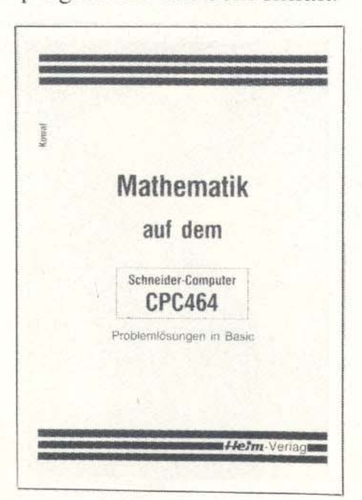
Vorbildlich ist die Dokumentation der einzelnen Programme. Schritt für Schritt wird das Programm erarbeitet und ist somit gerade für Einsteiger sehr ansprechend gestaltet.

In dem Kapitel „Veränderungen" bekommt man Ideen und Anleitungen zur Anpassung der Programme an eigene spezielle Bedürfnisse und kann so seine persönliche Programmiertechnik ausfeilen. Als Service bietet der MVG-Verlag alle abgedruckten Listings auf Kassette an, die zum Preis von ca. DM 38,– erhältlich ist.

Ein empfehlenswertes Buch, das vor allem durch die ausgezeichnete Dokumentation und Abarbeitung einzelner Programmierschritte besticht.

Kowal
Mathematik auf dem CPC 464
Heim Verlag 1985
258 Seiten
ISBN 3-923250-22-3
Preis: 49,– DM

Oft fällt es schwer, mathematische Probleme auf den CPC umzusetzen. Hier hilft das Mathematik-Buch des Heim Verlages, das sich ausschließlich mit Problemlösungen in Basic befaßt. Thematisch richtet sich dieses Buch an die Mittel- und Oberstufen. Schritt für Schritt werden die einzelnen Themen erarbeitet. Einige Beispielprogramme aus dem Inhalt:



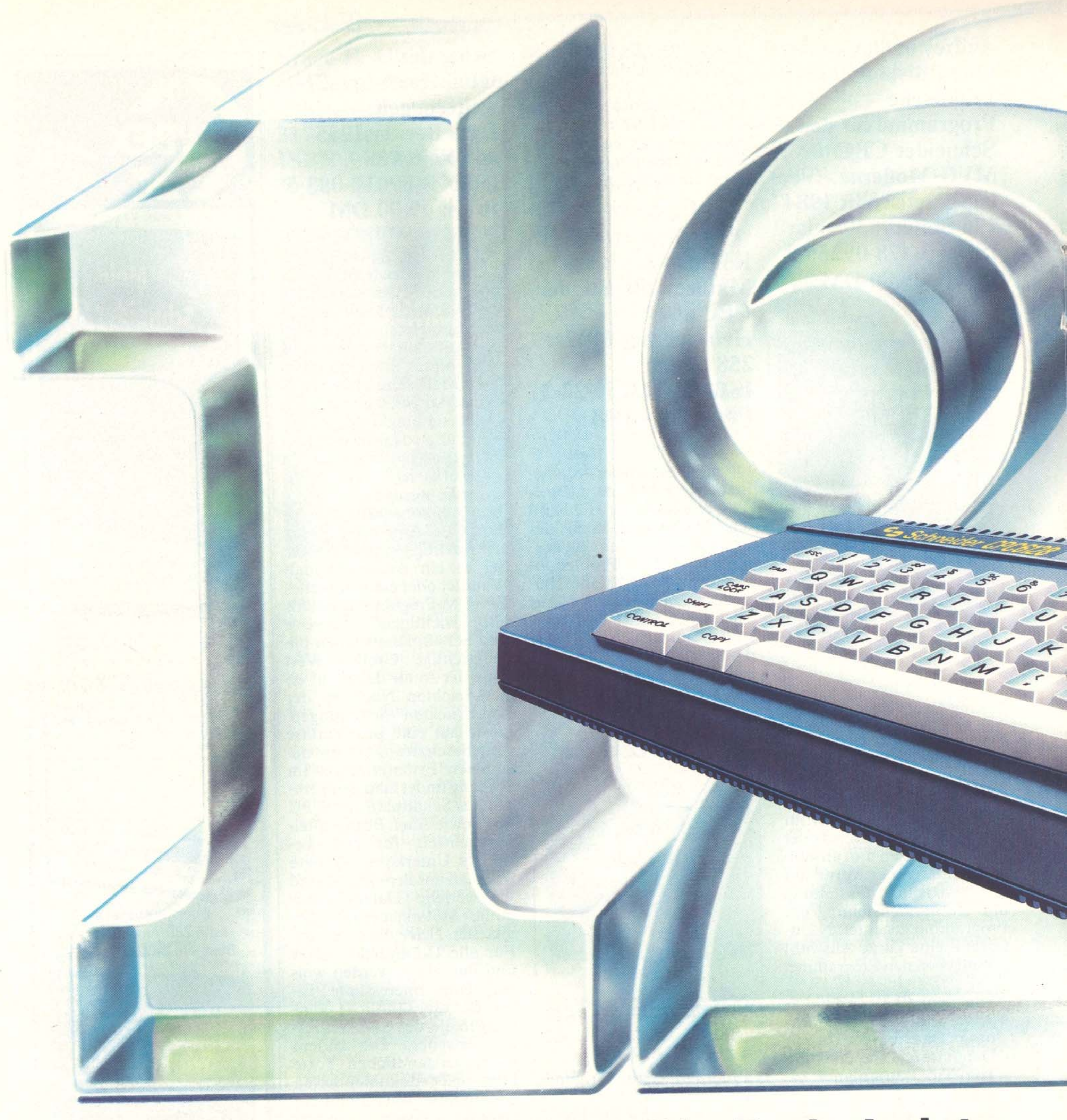
- Programme zur Trigonometrie
- Koordinatentransformationen
- Matrixalgebra
- Geometrie
- Zins, Dreisatz- und Prozentrechnung
- Logarithmen uvm.

Das übersichtliche und gut dokumentierte Buch hilft all denjenigen, die Probleme in Mathematik haben und Lösungswege auf dem Computer erarbeiten wollen. Lobenswert ist die gute Struktur der Beispielprogramme, die durchweg gut erläutert und mit Variablenlisten versehen sind.

Schüssler
CPC Hardware-Erweiterungen
Data Becker 1985
445 Seiten
ISBN 3-89011-083-5
Preis: 49,00 DM

Mit diesem Buch stellt Data Becker wieder einmal ansprechende Literatur für den CPC 464 vor. Für all diejenigen, die mehr aus Ihrem CPC „herausholen" möchten, sollte dieses Buch ein absolutes Muß sein. Hier findet man nützliche Tips zur Platinenherstellung und Adreßcodierung sowie Informationen und Hilfestellung zu Adapterkarten und Interfaces. Es werden u.a. Erweiterungen vorgestellt, auf die viele CPC-Anwender bereits sehnsüchtig gewartet haben. Sei es ein Analog-Digital-Wandler oder ein komplettes EPROM-Programmierboard; alle wichtigen Hinweise und Schaltpläne werden berücksichtigt. Großen Wert legte der Autor dabei auf einen leichten Nachbau der abgedruckten Schaltungen sowie auf eine problemlose Teilebeschaffung für die benötigten Erweiterungen. Im Anhang findet man den kompletten Schaltplan des CPC 464 sowie einen Bezugsquellennachweis für evtl. benötigte Unterlagen. Weitere Themen aus dem Inhalt sind u.a. serielle Datenübertragung, Motorsteuerung, Logiktester, Netzteilkarte u.v.m. Für alle Hobbyelektroniker und die, die es werden wollen, eine interessante Literatur, die sämtliche Einsatzgebiete des CPC wesentlich erweitert. Die zudem leicht verständlichen Texte und gut dokumentierten Schaltpläne und Platinenlayouts lassen keinen Wunsch offen.





Neu. Schneider CPC 6128. Die Preis-Leistungs
in der 128 K-Byte-Profiklasse.

128 K-Byte halten Ihnen den Rücken frei, auch bei aufwendigen und umfangreichen Programmen. Für ganze DM 1.598,–* gibt es den CPC 6128, die neue Profiklasse von Schneider, komplett mit Keyboard, integriertem 3"-Floppylaufwerk für blitzschnelles Laden und Abspeichern und Grün-Monitor. Mit diesem Gerät können Sie auch Meß- und Regelgeräte steuern, Daten und Befehle an andere Rechner weitergeben oder mit der entsprechenden Peripherie die zukünftigen Möglichkeiten der Telekommunikation nutzen.

* Unverbindliche Preisempfehlung
**eingetragene Warenzeichen der Digital Research Inc.

Technische Informationen für alle, die vergleichen wollen.
Schneider Personal Computer CPC 6128.
Lieferumfang: Keyboard mit integriertem 3"-Floppylaufwerk, Monitor, Softwarepaket aus 2 Disketten:

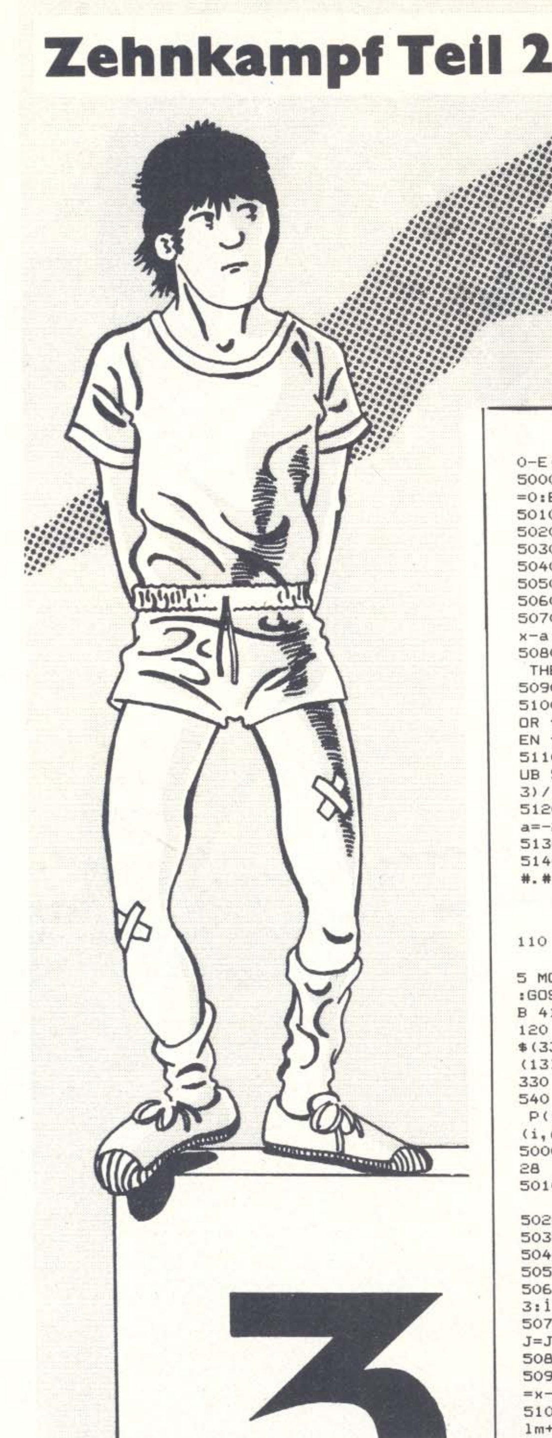
- CP/M 2.2** und CP/M Plus**, für den Zugriff auf ein nahezu unbegrenztes Software-Angebot
- LOGO in der Version Dr. LOGO** „Software des Jahres" 1984
- GSX** Grafikprogramm

Preis: mit Grün-Monitor **DM 1.598,–***
mit Farb-Monitor **DM 2.098.–***

Mikroprozessor: Z 80 A
Arbeitsspeicher: 128 K-RAM
Massenspeicher: 3"-Compact-Floppy-Disc
Bildschirm: monochrom oder Farbe, 20/40/80 Zeichen/Zeile, 25 Zeilen, 640 x 200 Bildpunkte
Ton: programmierbar, Mono-Lautsprecher eingebaut, 3 Kanäle Stereoton (über HiFi-Anlage)
Tastatur: Schreibmaschine, zusätzlicher Zehnerblock und Cursor-Steuertasten
Schnittstelle: parallel (Centronics-Komp.)
Software inkl.: CP/M 2.2, CP/M Plus, Dr. LOGO, BASIC, GSX Grafikprogramm


ssensation
Das Schneider CPC Computer-System.
Für jeden den richtigen Rechner:
Schneider CPC 464, ideal für den Einsteiger.
Komplettpreis für Keyboard mit
Datenrecorder und
Grün-Monitor DM 899,–*
Farb-Monitor DM 1.398,–*
Schneider CPC 664,
für den Anspruchsvollen.
Komplettpreis für Keyboard, mit
integriertem 3"-Diskettenlaufwerk und
Grün-Monitor DM 1.498,–*
Farb-Monitor DM 1.998,–*
Schneider CPC 6128, für den Profi.
Komplettpreis für Keyboard mit integriertem
3"-Diskettenlaufwerk und
Grün-Monitor DM 1.598,–*
Farb-Monitor 2.098,–*
Schneider
COMPUTER DIVISION
128K Colour Personal Computer
Disc Drive
Tasten–Nummern
CPC Farbtabelle
Schneider
CF2 Compact Floppy Disk
Schneider
Dateiverwaltung
Anfangen mit
beliebiger Taste

Zehnkampf Teil 2



Hier ist er nun, der zweite Teil des Programms Zehnkampf! Nach dem, hoffentlich fehlerfreien Abtippen der ersten fünf Disziplinen der vorhergehenden Ausgabe, ist das Simulationsprogramm mit allen 10 Disziplinen nunmehr komplett. Noch ein Hinweis zum Diskettenbetrieb: Sollten Sie das Programm auf dem CPC 464 mit Floppy eintippen, ist die Mergefix-Routine aus Heft 6/85 (S. 54) miteinzugeben. Besitzer des 664 brauchen dies nicht, da beim neuen CPC die Merge bzw. Chain-Merge-Befehle berichtigt wurden.

```
400 M        PROG.
5 MODE 1:GOSUB 40030:lk=3:dz=5:lm=206:ml
=26:GOSUB 41030
120 KEY 1,CHR$(13)+"merge "+CHR$(34)+CHR
$(33)+"110 m Huerden"+CHR$(13)+"run"+CHR
$(13)
330 CLS
340 GOSUB 1000
540 IF E(i,dz)>78.08 THEN P(i,dz)=0 ELSE
 P(i,dz)=CINT((100-E(i,dz))^2*0.5864-(10
0-E(i,dz))*12.836296)
5000 a=0:b=0:kl=468:lk=330:lm=351:zz=0:z
=0:EVERY 30 GOSUB 5140
5010 J=JOY(0):z=z+0.053
5020 IF J=8 THEN a=2:b=0
5030 IF J=4 THEN a=-2:b=0
5040 IF J=1 THEN a=0:b=2
5050 IF J=2 THEN a=0:b=-2
5060 PLOT x,y,0
5070 x=x+a:y=y+b:IF TEST(x,y)<>0 THEN x=
x-a:y=y-b
5080 PLOT x,y,2:IF x<>kl OR y<lk OR y>lm
 THEN 5010
5090 t=1:a=a*1.5
5100 z=z+0.043:J=JOY(0):IF J=t OR t+4=J
OR t+8=J THEN PLOT x,y,0:x=x+a:IF t=1 TH
EN t=2 ELSE t=1
5110 IF x=494 AND y<130 THEN EVERY 0 GOS
UB 5140:SOUND 1,200,4,7:z=zz+z+INT(RND*4
3)/100:RETURN
5120 IF x<170 THEN kl=170:lk=108:lm=130:
a=-2:FOR i=1 TO 300:NEXT:GOTO 5010
5130 PLOT x,y,2:GOTO 5100
5140 zz=zz+1:LOCATE 27,16:PRINT USING "#
#.#";zz:z=0:RETURN


110 M HUERDEN PROG.

5 MODE 1:SYMBOL 249,0,0,0,3,31,55,42,105
:GOSUB 40030:lk=3:dz=6:lm=260:ml=26:GOSU
B 41030
120 KEY 1,CHR$(13)+"merge "+CHR$(34)+CHR
$(33)+"Diskuswerfen"+CHR$(13)+"run"+CHR$
(13)
330 CLS
540 IF E(i,dz)>31.57 THEN P(i,dz)=0 ELSE
 P(i,dz)=CINT((32-E(i,dz))^2*3.097-(32-E
(i,dz))*0.15485)
5000 SYMBOL 250,56,56,18,124,144,44,68,1
28
5010 SYMBOL 251,56,56,16,56,16,40,196,2

5020 SYMBOL 252,56,56,20,56,80,24,40,8
5030 SYMBOL 253,56,56,20,56,16,46,64,128
5040 SYMBOL 254,0,0,0,12,31,44,33,223
5050 SYMBOL 255,56,56,16,24,20,232,4,2
5060 TAG:z=0:x=223:ch=250:t=1:s=0:a=0:b=
3:lm=0
5070 IF s>0 THEN s=s-s/10:J=JOY(0) ELSE
J=JOY(0)
5080 MOVE x,7:DRAWR a,b,2
5090 IF x>30 AND x<46 THEN a=3:b=0:s=0:x
=x-16:FOR i=1 TO 50:z=z+0.01:NEXT
5100 x=x-s:IF x<0 THEN x=223:a=0:b=3:lm=
lm+1
5110 IF lm=10 THEN 5360
5120 MOVE x,7:DRAWR a,b,3
5130 MOVE 30,21:PRINT " ";
5140 iii=iii+s/5:IF iii<2 THEN 5170 ELSE
 iii=0
5150 IF ch=252 THEN ch=250:GOTO 5170
5160 IF ch=251 THEN ch=252 ELSE ch=251
5170 PLOT 700,0,0:MOVE 30,21:PRINT CHR$(
ch);
5180 IF JOY(0)>15 THEN 5230
5190 z=z+0.02
5200 IF J<>t AND J<>t+4 AND J<>t+8 THEN
5070
5210 s=s+1.6:IF t=1 THEN t=2 ELSE t=1
5220 GOTO 5070
5230 ii=0:PLOT 700,0,0:MOVE 30,21:PRINT
CHR$(253);
5240 IF s>0 THEN s=s-s/14:J=JOY(0) ELSE
J=JOY(0)
5250 MOVE x,7:DRAWR a,b,2
5260 IF TEST(x-2,11)<>2 OR TEST(x-2,9)<>
2 OR TEST(x-2,7)<>2 THEN a=3:b=0:s=0:x=x
-16:FOR i=1 TO 50:z=z+0.01:NEXT
5270 x=x-s:IF x<0 THEN x=223:a=0:b=3:lm=
lm+1
5280 IF lm=10 THEN 5360
5290 MOVE x,7:DRAWR a,b,3
5300 IF (ii>6 AND ii<26) AND JOY(0)<15 T
HEN ii=25
5310 ii=ii+1:IF ii=6 THEN PLOT 700,0,0:M
OVE 30,21:PRINT " ";:MOVE 30,29:PRINT CH
R$(254);
5320 IF ii=26 THEN PLOT 700,0,0:MOVE 30,
29:PRINT " ";:MOVE 30,21:PRINT CHR$(255)
;
5330 IF ii=28 THEN 5070
5340 z=z+0.02
5350 GOTO 5240
5360 IF s>0 THEN s=s-s/10:J=JOY(0) ELSE
J=JOY(0)
5370 PLOT x,5,1:x=x-s
5380 PLOT x,5,3:IF x<46 THEN TAGOFF:z=z+
7+(RND*2+1)/100:RETURN
5390 PLOT 700,0,1:MOVE 30,21:PRINT " ";

5400 iii=iii+s/5:IF iii<2 THEN 5430 ELSE
 iii=0
5410 IF ch=252 THEN ch=250:GOTO 5430
5420 IF ch=251 THEN ch=252 ELSE ch=251
5430 PLOT 700,0,0:MOVE 30,21:PRINT CHR$(
ch);
5440 z=z+0.02
5450 IF J<>t AND J<>t+4 AND J<>t+8 THEN
5360
5460 s=s+1.6:IF t=1 THEN t=2 ELSE t=1
5470 GOTO 5360

DISKUSWERFEN  PROG.

5 MODE 1:GOSUB 40030:lk=3:dz=7:lm=206:ml
=-28:GOSUB 41030
120 KEY 1,CHR$(13)+"merge "+CHR$(34)+CHR
$(33)+"Stabhochsprung"+CHR$(13)+"run"+CH
R$(13)
330 CLS
340 GOSUB 2000
540 P(i,dz)=CINT((E(i,dz)-8.35)*20.86956
52)
5000 BORDER 11:INK 0,11:INK 1,0:INK 2,9:
INK 3,22
5010 SYMBOL 220,0,0,1,1,0,3,3,1
5020 SYMBOL 221,0,0,192,192,128,192,208,
224
5030 SYMBOL 222,1,1,1,1,1,2,2,4
5040 SYMBOL 223,192,192,64,64,64,64,64,6
4
5050 SYMBOL 224,0,0,0,0,0,1,2,4
5060 SYMBOL 225,0,0,224,224,64,240,232,2
32
5070 SYMBOL 226,8,0,0,0,1,1,1,2
5080 SYMBOL 227,224,224,160,160,32,32,64
,64
5090 SYMBOL 228,0,0,0,0,0,3,12,16
5100 SYMBOL 229,0,0,96,96,64,252,226,224
5110 SYMBOL 230,1,1,1,1,1,6,4,1
5120 SYMBOL 231,192,192,64,64,64,64,128
5130 SYMBOL 232,0,0,0,0,0,7,24
5140 SYMBOL 233,0,0,224,224,64,240,240,2
40
5150 SYMBOL 234,1,1,3,3,3,2,4,8
5160 SYMBOL 235,208,224,96,48,16,32,48
5170 SYMBOL 236,0,0,1,1,0,3,3,13
5180 SYMBOL 237,0,0,128,128,128,240,204,
194
5190 SYMBOL 238,0,0,0,1,3,2,2,6
5200 SYMBOL 239,224,224,224,160,32,28,4
5210 SYMBOL 240,0,0,3,35,25,7,1,1
5220 SYMBOL 241,0,0,0,0,0,248,196,192
5230 SYMBOL 242,0,0,1,3,2,2,2,6
5240 SYMBOL 243,224,224,160,48,28,2,2
5250 SYMBOL 244,0,0,14,14,5,3,7,9
5260 SYMBOL 245,0,0,0,0,192,32,128,192
5270 SYMBOL 246,8,8,8,0,0,1,2,2
5280 SYMBOL 247,224,224,160,160,144,16,1
6,24
5290 SYMBOL 248,0,0,6,6,4,15,19,33
5300 SYMBOL 249,0,0,0,0,0,32,192,192
5310 SYMBOL 250,64,0,0,0,0,1,2,2
5320 SYMBOL 251,224,224,160,160,144,16,1
6,24
5330 SYMBOL 252,0,0,1,1,0,1,3,2
5340 SYMBOL 253,0,0,192,192,142,240,192,
224
5350 SYMBOL 254,2,2,0,1,1,2,4,4
5360 SYMBOL 255,224,224,160,176,48,16,16
,24
5370 MOVE 0,82:PRINT "          ";:MOVE 0,9
8:PRINT "          ";
5380 FOR i=56 TO 66 STEP 2
5390 MOVE 0,i:DRAWR 639,0,2:NEXT:PLOT 70
0,0,1
5400 MOVE 94,66:DRAWR 0,-4,3:PLOT 700,0,
1
5410 x=14:y=98:iii=0:a=0:Ueb=0
5420 ch=220:ml=20:lm=10:kl=22:lk=10:GOSU
B 5590
5430 ch=224:ml=8:lm=18:kl=10:lk=16:GOSUB
 5590
5440 ch=228:ml=6:lm=16:kl=8:lk=14:GOSUB
5590
5450 ch=232:ml=20:lm=16:kl=20:lk=14:GOSU
B 5590
5460 ch=236:ml=26:lm=14:kl=28:lk=16:GOSU
B 5590
5470 IF iii=1 THEN 5500
5480 ch=240:ml=24:lm=12:kl=26:lk=14:GOSU
B 5590
5490 IF iii<>1 THEN 5440
5500 IF x>68 THEN 5855
5510 w=90
5520 MOVE 100,120:w=w-5:PRINT INT(w);:IF
 JOY(0)=16 AND w>6 THEN 5520
5530 MOVE 100,120::PRINT " ";
5540 w=w-RND*6:MOVE 100,120:PRINT INT(w)
;
5550 ch=244:ml=10:lm=20:kl=10:lk=18:GOSU
B 5590
5560 ch=248:ml=2:lm=18:kl=4:lk=16:GOSUB
```

```
5590
5570 ch=252:ml=600:lm=200:kl=600:lk=200:
GOSUB 5590
5580 GOTO 5640
5590 MOVE x,y:PRINT "  ";:MOVE x,y-16:PR
INT "  ";:x=x+INT(RND*2+1)
5600 MOVE x,y:PRINT CHR$(ch);CHR$(ch+1);

5610 MOVE x,y-16:PRINT CHR$(ch+2);CHR$(c
h+3);
5620 PLOT x+ml,y-lm,3:PLOT x+kl,y-lk:PLO
T 700,0,1
5630 GOSUB 5890:RETURN
5640 xx=x+32:a=a*20+3000-(ABS(45-w)*8):i
=-1:iii=0
5650 DEG
5660 i=i+1.4
5670 PLOT x,y,0:PLOT x+2,y:PLOT x+4,y
5680 x=(i*a/720-iii)+xx:y=(218*SIN(i))+9
3
5690 IF y<73 THEN 5760
5700 IF x<639 THEN 5740
5710 iii=iii+639:MODE 1
5720 FOR ik=59 TO 69 STEP 2:MOVE 0,ik:DR
AWR 639,0,2:NEXT:PLOT 700,0,1
5730 FOR ik=-2+iii/31.95 TO 34+iii/31.95
 STEP 4:LOCATE ik-(-3+iii/31.95),23:PRIN
T ik-(ik-(-2+iii/31.95))/2;:NEXT:TAG
5740 PLOT x,y,3:PLOT x+2,y:PLOT x+4,y
5750 GOTO 5660
5760 MOVE x,y:DRAWR 3,0,3:xx=x
5770 PLOT 700,0,1:MOVE 100,20:PRINT "Met
er :";
5780 IF iii=639 THEN lm=17 ELSE lm=37
5790 i=x-100
5800 i=i+2:SOUND 7,1000,2,7
5810 IF TEST(i,71)=3 OR TEST(i,69)=3 THE
N 5840
5820 PLOT i,71:MOVE 250,20:PRINT USING "
##.##";i*0.03125+lm;
5830 GOTO 5800
5840 FOR i=1 TO 500:NEXT:Vs(po(ii),v)=(x
x*0.03125+lm)+(RND*20+1)/100
5850 INK 1,11::GOSUB 4030:RETURN
5855 Vs(po(ii),v)=0:GOSUB 4030:RETURN
5860 '
5870 ' Anlauf Diskuswerfen
5880 '
5890 s=0:t=1
5900 J=JOY(0):IF J=16 AND(ch=236 OR ch=2
40) THEN iii=1
5910 a=a-1:IF a<0 THEN a=0
5920 s=s+a:IF s>290 THEN RETURN
5930 IF J<>t AND J<>t+4 AND J<>t+8 THEN
5900
5940 a=a+7
5950 IF t=1 THEN t=2 ELSE t=1
5960 GOTO 5900
```

STABHOCHSPRUNG PROG.

```
5 MODE 1:GOSUB 40030:lk=3:dz=8:lm=260:ml
=-28:GOSUB 41030
120 KEY 1,CHR$(13)+"merge "+CHR$(34)+CHR
$(33)+"Speerwerfen"+CHR$(13)+"run"+CHR$(
13)
330 CLS
340 BORDER 11:INK 0,11:INK 1,0:INK 2,3:I
NK 3,23:GOSUB 3000
540 P(i,dz)=CINT((INT(E(i,dz)*100)/100-1
.27)*288.915663)
5000 ' Einsetzprogramm Stabhochsprung
5010 '
5020 FOR i=66 TO 66+Hoe*20:PLOT 585,i,1:
NEXT:PLOT 585,i,3
5030 MOVE 568,66:DRAWR 0,3,1
5040 SYMBOL 220,0,0,24,24,16,57,126,188
5050 SYMBOL 221,0
5060 SYMBOL 222,56,56,40,40,40,200,144,2
4
5070 SYMBOL 223,C,0,0,0,0,0,1,1

5080 SYMBOL 224,56,56,100,66,68,136,8
5090 SYMBOL 225,0,0,0,0,0,3,2
5100 SYMBOL 226,56,56,40,68,196,2,2,3
5110 SYMBOL 227,0,0,24,24,16,124,186,185
5120 SYMBOL 228,56,56,60,36,36,72,136,12
8
5130 SYMBOL 229,0,0,24,152,144,253,58,56
5140 SYMBOL 230,56,56,24,24,120,80,16,24
5150 SYMBOL 231,16,32,88,89,81,126,56,56
5160 SYMBOL 232,56,56,40,68,130,2,3
5170 SYMBOL 233,32,64,88,88,81,127,56,56
5180 SYMBOL 234,56,56,30,60,48,16,32,32
5190 SYMBOL 235,128,128,90,90,82,124,56,
56
5200 SYMBOL 236,56,56,62,18,18,34,67,64
5210 SYMBOL 237,128,130,90,90,82,124,56,
56
5220 SYMBOL 238,56,56,60,52,36,68,134,12
8
5230 SYMBOL 239,64,32,38,22,20,15,3,1
5240 SYMBOL 240,0,64,64,64,192,0,128,192
5250 SYMBOL 241,224,120,72,72,44,32,48
5260 SYMBOL 242,0,68,34,18,26,108,127,7
5270 SYMBOL 243,16,16,40,120,64,64,192,1
28
5280 SYMBOL 244,0,2,3,6,6,67,35,39
5290 SYMBOL 245,46,220,248
5300 SYMBOL 246,0,0,12,4,4,4,18,19
5310 SYMBOL 247,19,23,30,28,216,240
5320 SYMBOL 248,0,0,0,0,20,20,20,20
5330 SYMBOL 249,20,92,92,92,62,218,244,1
92
5340 SYMBOL 250,0,56,48,48,48,48,120,184
5350 SYMBOL 251,184,240,248,40,116,116,8
5360 SYMBOL 252,0,3,6,12,24,112,112,224
5370 SYMBOL 253,224,192,160,208,208
5380 SYMBOL 254,0,0,0,0,0,7,255,55
5390 SYMBOL 255,0,0,0,0,192,252,196
5400 PLOT 700,100,1
5410 x=-20:y=96:ch=221:t=1:s=0
5420 MOVE x+16,y:PRINT CHR$(220);
5430 MOVE x,y-16:PRINT CHR$(ch);CHR$(ch+
1);
5440 MOVE x+8,y-16:DRAWR 80,19,3:PLOT -1
0,-10,1
5450 IF JOY(0)<>16 THEN 5450
5460 FOR i=1 TO 500:NEXT
5470 IF s>0 THEN s=s-0.12:J=JOY(0) ELSE
J=JOY(0)
5480 MOVE x+16,y:PRINT " ";:MOVE x,y-16:
PRINT "  ";:MOVE x+8,y-16:DRAWR 80,19,0:
PLOT -10,-10,1
5490 iii=iii+s/5:IF iii<2 THEN 5530 ELSE
 iii=0
5500 IF ch=225 THEN ch=221:GOTO 5530

5510 IF ch=223 THEN ch=225
5520 IF ch=221 THEN ch=223
5530 x=x+s:MOVE x+16,y:PRINT CHR$(220);:
MOVE x,y-16:PRINT CHR$(ch);CHR$(ch+1);:M
OVE x+8,y-16:DRAWR 80,19,3:PLOT -10,-10,
1
5540 IF x>556 THEN MOVE x+16,y:PRINT " "
;:MOVE x,y-16:PRINT "  ";:MOVE x+8,y-16:
DRAWR 80,19,0:PLOT -10,-10,1:RETURN
5550 IF JOY(0)=16 THEN GOTO 5590
5560 IF J<>t AND J<>t+4 AND J<>t+8 THEN
5470
5570 s=s+1.6:IF t=1 THEN t=2 ELSE t=1

5580 GOTO 5470
5590 MOVE x+16,y:PRINT " ";:MOVE x,y-16:
PRINT "  ";:MOVE x+8,y-16:DRAWR 80,19,0:
PLOT -10,-10,1
5600 x=x+18:MOVE x,y:PRINT CHR$(227);:MO
VE x,y-16:PRINT CHR$(228);:MOVE x-8,y-16
:DRAWR 80,0,3:PLOT -10,-10,1
5610 FOR i=1 TO 50:NEXT:MOVE x,y:PRINT "
 ";:MOVE x,y-16:PRINT " ";:MOVE x-8,y-16
:DRAWR 80,0,0:PLOT -10,-10,1
5620 ch=221:IF x>489 THEN 6630

5630 s=s-1.6:IF s<0 THEN x=x-2:GOTO 6630

5640 MOVE x,y:PRINT " ";:MOVE x-16,y-16:
PRINT "  ";:MOVE x-8,y-4:DRAWR 80,-25,0:
PLOT -10,-10,1
5650 iii=iii+s/5:IF iii<2 THEN 5690 ELSE
 iii=0
5660 IF ch=225 THEN ch=221:GOTO 5690
5670 IF ch=223 THEN ch=225
5680 IF ch=221 THEN ch=223
5690 x=x+s:IF x>489 THEN x=491:GOTO 5720

5700 MOVE x,y:PRINT CHR$(229);:MOVE x-16
,y-16:PRINT CHR$(ch);CHR$(ch+1);:MOVE x-
8,y-4:DRAWR 80,-25,3:PLOT -10,-10,1
5710 GOTO 5630
5720 MOVE x,y:PRINT CHR$(229);:MOVE x,y-
16:PRINT CHR$(230);:MOVE x-8,y-4:DRAWR 8
0,-25,3:PLOT -10,-10,1
5730 s=s-1:GOSUB 6650
5740 '
5750 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x-8,y-4:DRAWR 80,-25,0:PLOT
 -10,-10,1
5760 x=x+2:MOVE x,y:PRINT CHR$(231);:MOV
E x-4,y-24:DRAWR 3,0:PLOT x-4,y-26:MOVE
x,y-16:PRINT CHR$(232);:MOVE x-8,y+4:DRA
WR 80,-34,3:PLOT -10,-10,1
5770 s=s-1.4:GOSUB 6650
5780 '
5790 MOVE x,y:PRINT " ";:MOVE x-4,y-24:D
RAWR 3,0,0:PLOT x-4,y-26,0:MOVE x,y-16:P
RINT " ";:MOVE x-8,y+4:DRAWR 80,-34:PLOT
 -10,-10,1
5800 MOVE x,y:PRINT CHR$(233);:MOVE x,y-
16:PRINT CHR$(234);:MOVE x-4,y+2:DRAWR 3
2,-10,3:DRAW 566,66:PLOT -10,-10,1
5810 s=s-1.8:GOSUB 6650
5820 '
5830 MOVE x-4,y+2:DRAWR 32,-10,0:DRAW 56
6,66:PLOT -10,-10,1
5840 MOVE x,y:PRINT CHR$(235);:MOVE x,y-
16:PRINT CHR$(236);:MOVE x-4,y+2:DRAWR 3
6,-6,3:DRAW 566,66:PLOT -10,-10,1
5850 s=s-2.2:GOSUB 6650
5860 '
5870 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x-4,y+2:DRAWR 36,-6,0:DRAW
566,66:PLOT -10,-10,1
5880 y=y+8:x=x+10:MOVE x,y:PRINT CHR$(23
7);:MOVE x,y-16:PRINT CHR$(238);:MOVE x-
4,y+2:DRAWR 32,-4,3:DRAWR 24,-14:DRAW 56
6,66:PLOT -10,-10,1
5890 s=s-2.6:GOSUB 6650
5900 '
5910 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x-4,y+2:DRAWR 32,-4,0:DRAWR
 24,-14:DRAW 566,66:PLOT -10,-10,1
5920 y=y+6:x=x+6:MOVE x,y:PRINT CHR$(239
);:MOVE x+16,y:PRINT CHR$(240);:MOVE x+1
6,y-16:PRINT CHR$(241);:MOVE x-2,y+2:DRA
WR 26,-4,3:DRAWR 8,-6:DRAWR 10,-8:DRAW 5
66,66:PLOT -10,-10,1
5930 s=s-3:GOSUB 6650
5940 '
5950 MOVE x,y:PRINT " ";:MOVE x+16,y:PRI
NT " ";:MOVE x+16,y-16:PRINT " ";:MOVE x
-2,y+2:DRAWR 26,-4,0:DRAWR 8,-6:DRAWR 10
,-8:DRAW 566,66:PLOT -10,-10,1
5960 y=y+12:x=x+12:MOVE x,y:PRINT CHR$(2
42);CHR$(243);:MOVE x-2,y+2:DRAWR 24,-8,
3:DRAWR 12,-10:DRAWR 10,-22:DRAW 566,66:
PLOT -10,-10,1
5970 s=s-3.4:GOSUB 6650
5980 '
5990 MOVE x,y:PRINT "  ";:MOVE x-2,y+2:D
RAWR 24,-8,0:DRAWR 12,-10:DRAWR 10,-22:D
RAW 566,66:PLOT -10,-10,1
6000 y=y+16:x=x+10:MOVE x,y:PRINT CHR$(2
44);:MOVE x,y-16:PRINT CHR$(245);:MOVE x
-2,y-6:DRAWR 20,-10,3:DRAWR 10,-12:DRAWR
 8,-22:DRAW 566,66:PLOT -10,-10,1
6010 s=s-3.8:GOSUB 6650
6020 '
6030 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x-2,y-6:DRAWR 20,-10,0:DRAW
R 10,-12:DRAWR 8,-22:DRAW 566,66:PLOT -1
0,-10,1
6040 y=y+10:x=x+10:MOVE x,y:PRINT CHR$(2
46);:MOVE x,y-16:PRINT CHR$(247);:MOVE x
,y-6:DRAWR 12,-12,3:DRAWR 8,-12:DRAWR 6,
-22:DRAW 566,66:PLOT -10,-10,1
6050 s=s-4.2:GOSUB 6650
6060 '
6070 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x,y-6:DRAWR 12,-12,0:DRAWR
8,-12:DRAWR 6,-22:DRAW 566,66:PLOT -10,-
10,1
6080 y=y+14:x=x+16:MOVE x,y:PRINT CHR$(2
48);:MOVE x,y-16:PRINT CHR$(249);:MOVE x
+4,y-16:DRAWR 6,-24,3:DRAWR -2,-44:DRAW
566,66:PLOT -10,-10,1
6090 s=s-4.6:GOSUB 6650
6100 '
6110 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x+4,y-16:DRAWR 6,-24,0:DRAW
R -2,-44:DRAW 566,66:PLOT -10,-10,1
6120 x=x+4
6130 ml=s+9:i=-1:IF ml<0 THEN ml=0
6140 i=i+1
6150 MOVE x,y+i-1:PRINT " ";:MOVE x,y-16
-1:PRINT " ";
6160 MOVE x,y+i:PRINT CHR$(250);:MOVE x,
y+i-16:PRINT CHR$(251);:MOVE x+6,y-16:DR
AWR -2,-16,3:DRAWR -2,-16:DRAW 566,66:PL
OT -10,-10,1
6170 IF JOY(0)<>0 THEN w=90:GOTO 6240
6180 IF i<ml THEN 6140
6190 i=i-1
6200 MOVE x,y+i-1:PRINT " ";:MOVE x,y-16
-1:PRINT " ";
6210 MOVE x,y+i:PRINT CHR$(250);:MOVE x,
y+i-16:PRINT CHR$(251);:MOVE x+6,y-16:DR
AWR -2,-16,3:DRAWR -2,-16:DRAW 566,66:PL
OT -10,-10,1
6220 IF JOY(0)<>0 THEN 6240
6230 IF i>0 THEN 6190
6240 w=90
6250 MOVE 555,210:w=w-5:PRINT w;:IF JOY(
0)<>0 AND w>5 THEN 6250
6260 MOVE 555,210:w=w-CINT(RND*5):PRINT
w;
6270 s=(10*(s+10+(20-(ABS(45-w))/2.25)))
/30
6280 MOVE x,y+i:PRINT " ";:MOVE x,y+i-16
:PRINT " ";
6290 DEG
6300 xx=x:yy=y:FOR n=0 TO 35 STEP 8
6310 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT " ";
6320 xx=n*0.2+x:yy=s*SIN(n)+y+i
6330 MOVE xx,yy:PRINT CHR$(252);:MOVE xx
,yy-16:PRINT CHR$(253);:MOVE x+6,y-16:DR
AWR -2,-16,3:DRAWR -2,-16:DRAW 566,66:PL
OT -10,-10,1
6340 NEXT
6350 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT " ";:MOVE x+6,y-16:DRAWR -2,-16,3
:DRAWR -2,-16:DRAW 566,66:PLOT -10,-10,1

6360 FOR n=35 TO 145 STEP 9
6370 MOVE xx,yy:PRINT "  ";
6380 xx=n*0.2+x:yy=s*SIN(n)+y+i
6390 MOVE xx,yy:PRINT CHR$(254);CHR$(255
);
6400 NEXT
6410 SYMBOL 220,0,0,0,12,204,36,28,14:SY
MBOL 221,6,7,3,1:SYMBOL 222,0,0,128,128,
248,8
6420 MOVE xx,yy:PRINT "  ";
6430 FOR n=145 TO 180 STEP 8
6440 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT "  ";
6450 xx=n*0.2+x:yy=s*SIN(n)+y+i
6460 MOVE xx,yy:PRINT CHR$(220);:MOVE xx
,yy-16:PRINT CHR$(221);CHR$(222);
6470 NEXT
6480 SYMBOL 223,0,0,49,50,20,56,112,112:
SYMBOL 224,224,224,64,64,120,8
6490 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT "  ";:xx=xx+4
6500 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT " ";
6510 yy=yy-2:IF yy<120 THEN 6540
6520 MOVE xx,yy:PRINT CHR$(223);:MOVE xx
,yy-16:PRINT CHR$(224);
6530 GOTO 6500
6540 SYMBOL 225,0,0,0,32,32,16,15,15:SYM
BOL 226,0,0,0,64,128,176,240,128
6550 MOVE xx,yy:PRINT " ";:MOVE xx,yy-16
:PRINT " ";:xx=xx+4:yy=yy-10
6560 MOVE xx,yy:PRINT "  ";
6570 yy=yy-2:IF yy<89 THEN yy=89:GOTO 66
00
6580 MOVE xx,yy:PRINT CHR$(225);:MOVE xx
```

```
+16,yy:PRINT CHR$(226);
6590 GOTO 6560
6600 MOVE xx,yy:PRINT CHR$(225);:MOVE xx
+16,yy:PRINT CHR$(226);:FOR i=1 TO 1000:
NEXT
6610 MOVE x+6,y-16:DRAWR -2,-16,0:DRAWR
-2,-16:DRAW 566,66:PLOT -10,-10,1
6620 MOVE xx,yy:PRINT "  ";:x=xx:y=yy:RE
TURN
6630 x=x+2:MOVE x,y:PRINT CHR$(229);:MOV
E x-16,y-16:PRINT " ";CHR$(230);:MOVE x-
8,y-4:DRAWR 80,-25,3:PLOT -10,-10,1:FOR
i=1 TO 1000:NEXT
6640 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:MOVE x-8,y-4:DRAWR 80,-25,0:PLOT
 -10,-10,1:RETURN
6650 FOR i=1 TO (35-s)^1.2:NEXT:RETURN
```

SPEERWERFEN PROG.

```
5 MODE 1:GOSUB 40030:lk=3:dz=9:lm=314:ml
=-28:GOSUB 41030
120 KEY 1,CHR$(13)+"merge "+CHR$(34)+CHR
$(33)+"1500 m"+CHR$(13)+"run"+CHR$(13)
330 CLS
340 GOSUB 2000
540 P(i,dz)=CINT((E(i,dz)-5.4)*13.786764
7)
5000 BORDER 11:INK 0,11:INK 1,0:INK 2,9:
INK 3,24
5005 SYMBOL 220,0,0,24,26,18,124,120,184

5010 SYMBOL 221,56,56,40,40,40,200,144,2
4
5020 SYMBOL 222,56,56,100,66,68,136,8
5030 SYMBOL 223,56,56,40,76,196,2,2,3
5040 SYMBOL 224,16,8,24,24,16,56,60,56
5050 SYMBOL 225,56,56,110,66,66,131
5060 SYMBOL 226,0,128,152,152,80,124,58,
58
5070 SYMBOL 227,58,58,40,36,68,130,2,3
5080 SYMBOL 228,0,0,24,24,16,253,58,56
5090 SYMBOL 229,56,56,24,20,36,68,134,12
8
5100 SYMBOL 230,0,0,24,24,16,254,57,56
5110 SYMBOL 231,56,56,40,44,68,130,2,3
5120 SYMBOL 232,0,0,48,48,32,248,116,58
5130 SYMBOL 233,57,56,40,36,36,36,40,40
5140 SYMBOL 234,57,56,40,68,132,2,3
5150 SYMBOL 235,0,0,96,96,64,248,118,113
5160 SYMBOL 236,56,56,40,36,66,130,2,3
5170 SYMBOL 237,0,0,96,97,70,248,112,112
5180 SYMBOL 238,56,56,40,36,34,18,19,24
5190 SYMBOL 239,0,0,3,3,2,15,115,131
5200 SYMBOL 240,0,0,0,0,8,240,128,128
5210 SYMBOL 241,1,1,1,1,1,6,8,8
5220 SYMBOL 242,192,192,64,48,16,8,6
5230 SYMBOL 243,0,0,192,192,128,192,224,
113
5240 SYMBOL 244,62,56,40,68,132,2,2,3
5250 SYMBOL 245,2,4,12,12,8,29,30,56
5260 SYMBOL 246,56,56,40,36,36,66,130,13
1
5270 SYMBOL 247,0,0,3,3,2,31,39,78
5280 SYMBOL 248,14,14,10,26,50,65,129,12
9
5290 SYMBOL 249,0,0,24,24,16,124,186,186
5300 SYMBOL 250,186,60,40,40,68,132,2,3
5310 SYMBOL 251,0
5320 SYMBOL 252,0,0,0,0,0,0,1,1
5330 SYMBOL 253,0,0,0,0,0,3,2
5340 FOR i=56 TO 66 STEP 2
5350 MOVE 0,i:DRAWR 639,0,2:NEXT
5360 MOVE 600,66:DRAWR 0,-4,3

5380 x=-20:y=98:ch=221:t=1:s=0
5390 IF s>0 THEN s=s-0.12:J=JOY(0) ELSE
J=JOY(0)
5400 iii=iii+s/5:IF iii<2 THEN 5440 ELSE
 iii=0
5410 IF ch=223 THEN ch=221:GOTO 5440
5420 IF ch=222 THEN ch=223
5430 IF ch=221 THEN ch=222
5440 MOVE x,y:PRINT " ";:MOVE x-16,y-16:
PRINT "  ";:MOVE x-16,y-4:DRAWR 50,0,0:P
LOT -10,-10,1
5450 x=x+s:MOVE x,y:PRINT CHR$(220);:MOV
E x-16,y-16:PRINT CHR$(ch+30);CHR$(ch);:
MOVE x-16,y-4:DRAWR 50,0,3:PLOT -10,-10,
1
5460 IF x>585 THEN 6320
5470 IF JOY(0)=16 THEN 5520
5480 FOR i=1 TO 10:NEXT
5490 IF J<>t AND J<>t+4 AND J<>t+8 THEN
5390
5500 s=s+1.6:IF t=1 THEN t=2 ELSE t=1

5510 GOTO 5390
5520 s=s+(CINT(RND*12)/100):w=90
5530 MOVE x,120:w=w-5:PRINT INT(w);:IF J
OY(0)=16 AND w>6 THEN 5530
5540 w=w-CINT(RND*5):MOVE x,120:PRINT w;

5550 MOVE x-16,y-16:PRINT" ";:MOVE x-16,
y-4:DRAWR 50,0,0:PLOT -10,-10,1
5560 ch=224:ml=-16:lm=-2:kl=50:lk=0:GOSU
B 6330:MOVE x-2,y-28:DRAWR 0,-2
5570 FOR i=1 TO 1:NEXT
5580 MOVE x-2,y-28:DRAWR 0,-2,0:MOVE x-1
6,y+2:DRAWR 50,0:PLOT -10,-10,1
5590 ch=226:y=y+2:ml=-24:lm=4:kl=46:lk=4
:GOSUB 6330:MOVE x-2,y-28:DRAWR 0,-2
5600 FOR i=1 TO 1:NEXT
5610 MOVE x-2,y-28:DRAWR 0,-2,0:MOVE x-2
4,y-4:DRAWR 46,4,0:PLOT -10,-10,1
5620 ch=228:y=y-2:ml=-26:lm=6:kl=44:lk=6
:GOSUB 6330:MOVE x-4,y-6:DRAWR 0,-2:PLOT
 x-2,y-8
5630 FOR i=1 TO 1:NEXT
5640 MOVE x-4,y-6:DRAWR 0,-2,0:PLOT x-2,
y-8:MOVE x-26,y-6:DRAWR 44,6:PLOT x,y-30
:PLOT -10,-10,1
5650 ch=230:y=y+2:ml=-28:lm=6:kl=42:lk=8
:GOSUB 6330:MOVE x-6,y-4:DRAWR 6,-6:MOVE
 x-4,y-28:DRAWR 0,-2:PLOT x-2,y-26
5660 FOR i=1 TO 1:NEXT
5670 MOVE x-6,y-4:DRAWR 6,-6,0:MOVE x-4,
y-28:DRAWR 0,-2:PLOT x-2,y-26:MOVE x-28,
y-6:DRAWR 42,8:PLOT -10,-10,1
5680 ch=232:y=y-2:ml=-30:lm=10:kl=40:lk=
10:GOSUB 6330:MOVE x-8,y-6:DRAWR 8,-3
5690 FOR i=1 TO 1:NEXT
5700 MOVE x-8,y-6:DRAWR 8,-3,0:MOVE x-30
,y-10:DRAWR 40,10:PLOT -10,-10,1
5710 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:x=x+2:MOVE x,y:PRINT CHR$(232);:
MOVE x,y-16:PRINT CHR$(234);:MOVE x-32,y
-10:DRAWR 40,10,3:PLOT 700,0,1
5720 MOVE x-8,y-6:DRAWR 8,-3:MOVE x-4,y-
28:DRAWR 0,-2:PLOT x-2,y-26
5730 FOR i=1 TO 1:NEXT
5740 MOVE x-8,y-6:DRAWR 8,-3,0:MOVE x-4,
y-28:DRAWR 0,-2:PLOT x-2,y-26:MOVE x-32,
y-10:DRAWR 40,10:PLOT 700,0,1
5750 ch=235:ml=-30:lm=12:kl=38:lk=12:GOS
UB 6330:MOVE x-2,y-26:DRAWR 0,-2:MOVE x-
8,y-6:DRAWR 8,-3
5760 FOR i=1 TO 1:NEXT
5770 MOVE x-2,y-26:DRAWR 0,-2,0:MOVE x-8
,y-6:DRAWR 8,-3:MOVE x-30,y-12:DRAWR 38,
12:PLOT -10,-10,1
5780 ch=237:ml=-22:lm=20:kl=28:lk=22:GOS
UB 6330:PLOT x-10,y-14:MOVE x-8,y-12:DRA
WR 5,0:PLOT x-2,y-10
5790 FOR i=1 TO 1:NEXT
5800 PLOT x-10,y-14,0:MOVE x-8,y-12:DRAW
R 5,0:PLOT x-2,y-10:MOVE x-22,y-20:DRAWR
 28,22:PLOT -10,-10,1
5810 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:x=x+4
5820 MOVE x-10,y:PRINT CHR$(239);CHR$(24
0);:MOVE x-10,y-16:PRINT CHR$(241);CHR$(
242);
5830 MOVE x-22,y-20:DRAWR 28,22,3:PLOT 7
00,0,1
5840 FOR i=1 TO 1:NEXT
5850 MOVE x-10,y:PRINT "  ";:MOVE x-10,y
-16:PRINT "  ";:MOVE x-22,y-20:DRAWR 28,
22,0:PLOT 700,0,1
5860 ch=243:ml=-24:lm=20:kl=30:lk=20:GOS
UB 6330:MOVE x-4,y-28:DRAWR 0,-2:PLOT x-
2,y-26:MOVE x-12,y-14:DRAWR 8,0:MOVE x-4
,y-12:DRAWR 2,0
5870 FOR i=1 TO 1:NEXT
5880 MOVE x-4,y-28:DRAWR 0,-2,0:PLOT x-2
,y-26:MOVE x-12,y-14:DRAWR 8,0:MOVE x-4,
y-12:DRAWR 2,0:MOVE x-24,y-20:DRAWR 30,2
0:PLOT 700,0,1
5890 ch=245:ml=-4:lm=6:kl=32:lk=18:GOSUB
 6330
5900 FOR i=1 TO 1:NEXT
5910 MOVE x-4,y-6:DRAWR 32,18,0:PLOT 700
,0,1
5920 ch=247:ml=6:lm=8:kl=34:lk=16:GOSUB
6330:MOVE x+16,y-10:DRAWR 6,6:PLOT x+16,
y-30
5930 MOVE x+6,y-8:DRAWR 34,16,0
5940 IF x>585 THEN 6320
5950 DEG:y=89:xx=x+36:a=2*(s^2+2500-(ABS
(45-w)*20)):i=-1:iii=0:n=0
5960 FOR xr=1 TO 10:NEXT:i=i+2
5970 MOVE x,y:DRAWR -kl,yy,0
5980 lk=y:x=(i*a/720-iii)+xx
5990 y=a/20-81:IF y<219 THEN y=219
6000 y=(y*SIN(i))+102:IF y<lk THEN n=1

6010 IF n<>1 THEN 6110
6020 IF y<370 THEN kl=50
6030 IF y<320 THEN kl=46
6040 IF y<280 THEN kl=42
6050 IF y<250 THEN kl=38
6060 IF y<220 THEN kl=34
6070 IF y<200 THEN kl=30
6080 IF y<180 THEN kl=26
6090 IF y<160 THEN kl=22
6100 yy=50-kl:GOTO 6170
6110 IF y<370 THEN kl=50
6120 IF y<320 THEN kl=46
6130 IF y<270 THEN kl=42
6140 IF y<220 THEN kl=38
6150 IF y<170 THEN kl=34
6160 yy=(50-kl)*-1
6170 IF x<639 THEN 6210
6180 iii=iii+639:MODE 1
6190 FOR ik=59 TO 69 STEP 2:MOVE 0,ik:DR
AWR 639,0,2:NEXT:PLOT 700,0,1
6200 FOR ik=-2+iii/31.95 TO 34+iii/31.95
 STEP 4:LOCATE ik-(-3+iii/31.95),23:PRIN
T ik-(ik-(-2+iii/31.95))/2;:NEXT:TAG:GOT
O 5960
6210 IF y>73 THEN MOVE x,y:DRAWR -kl,yy,
3:GOTO 5960
6220 MOVE x,71:DRAWR -kl,yy,3:xx=x
6230 PLOT 700,0,1:MOVE 100,20:PRINT "Met
er :";
6234 IF iii=2556 THEN lm=77
6235 IF iii=1917 THEN lm=57
6240 IF iii=1278 THEN lm=37
6245 IF iii=639 THEN lm=17
6250 i=x-100
6260 i=i+2:SOUND 7,1000,2,7
6270 IF TEST(i,71)=3 OR TEST(i,69)=3 THE
N 6300
6280 PLOT i,71:MOVE 250,20:PRINT USING "
##.##";i*0.03125+lm;
6290 GOTO 6260
6300 FOR i=1 TO 500:NEXT:Vs(po(ii),v)=(x
x*0.03125+lm)
6310 INK 1,11::GOSUB 4030:RETURN
6320 Vs(po(ii),v)=0:MOVE 140+v*128,395-i
i*22:PRINT USING "##.##";Vs(po(ii),v);:P
RINT " m";:FOR i=1 TO 1000:NEXT:RETURN
6330 MOVE x,y:PRINT " ";:MOVE x,y-16:PRI
NT " ";:x=x+CINT(RND*2+8)
6340 MOVE x,y:PRINT CHR$(ch);:MOVE x,y-1
6:PRINT CHR$(ch+1);
6350 MOVE x+ml,y-lm:DRAWR kl,lk,3:PLOT 7
00,0,1
6360 RETURN
```

1500 M PROG.

```
1 SYMBOL 220,102,0,102,102,102,102,62
2 da=PEEK(40300)
5 MODE 1:GOSUB 40030:lk=3:dz=10:lm=368:m
l=-28:GOSUB 41030
120 KEY 1,"1"
130 KEY 2,"2"
203 FOR i=1 TO 16:IF i>LEN(GW$) THEN POK
E 40259+i,32 ELSE POKE 40259+i,ASC(MID$(
GW$,i,1))
204 NEXT
207 FOR i=1 TO 10:PU(i)=HE(i):NEXT:kl=40
438:GOSUB 245
209 a=INT(HGP/256):b=HGP-a*256:POKE 4046
0,a:POKE 40461,b
211 FOR i=1 TO 10:PU(i)=HP(i):NEXT:kl=40
478:GOSUB 257
235 NEXT:GOTO 8000
330 CLS
340 GOSUB 1000
350 GOTO 201
540 IF E(i,dz)>398 THEN P(i,dz)=0 ELSE P
(i,dz)=CINT((401-E(i,dz))^1.6*0.2636-(40
1-E(i,dz))*0.279113612)
5000 lm=0:a=0:b=0:zz=0:z=0:EVERY 19 GOSU
B 5110
5010 LOCATE 28,15:PRINT "0.0  s"
5020 J=JOY(0):z=z+0.1
5030 IF J=8 THEN a=2:b=0
5040 IF J=4 THEN a=-2:b=0
5050 IF J=1 THEN a=0:b=2
5060 IF J=2 THEN a=0:b=-2
5070 IF x=494 AND y<130 THEN lm=lm+1:LOC
ATE 26,15:PRINT USING "###.#";zz+z
5080 IF lm=4 THEN EVERY 0 GOSUB 5110:SOU
ND 7,2000,4,7:PLOT x,y,0:z=zz+z+INT(RND*
10)/100:RETURN
5090 PLOT x,y,0:x=x+a:y=y+b:IF TEST(x,y)
<>0 THEN x=x-a:y=y-b
5100 PLOT x,y,2:GOTO 5020
5110 zz=zz+1:LOCATE 26,16:PRINT USING "#
##.##";zz:z=0:RETURN
8000 FOR i=1 TO 4
8010 IF GP(i)<=HGP THEN 8030
8020 da=1:GW$=N$(i):HGP=GP(i):FOR ii=1 T
O 10:HE(ii)=E(i,ii):HP(ii)=P(i,ii):NEXT
8030 NEXT
8040 INK 3,0,26:MODE 1:LOCATE 1,4:PRINT
"Dr";CHR$(220);"cken Sie"
8050 IF da=1 THEN PEN 3
8060 LOCATE 12,7:PRINT "(D)  fuer Datens
peicherung":PEN 1
8070 LOCATE 12,9:PRINT "(E)  fuer Ergebn
istabelle"
8080 LOCATE 12,11:PRINT "(S)  fuer ein n
eues Spiel"
8090 LOCATE 12,13:PRINT "(W)  fuer Weltr
ekordtabelle"
8100 IF INKEY(61)=0 THEN GOSUB 8140:GOTO
 8040
8110 IF INKEY(58)=0 THEN GOSUB 30000:GOT
O 8040
8120 IF INKEY(60)=0 THEN  CLS:PRINT "Bit
te 'ENTER' vom Zehnerblock u. gleich- ze
itig 'CTRL' dr";CHR$(220);"cken und neu
Laden !!!":LOCATE 1,15:END
8130 IF INKEY(59)=0 THEN  GOSUB 31000:GO
TO 8040 ELSE 8100
8140 CLS:LOCATE 10,2:PRINT "DATENSPEICHE
RUNG"
8150 LOCATE 2,5:PRINT "- Legen Sie die C
assette ein, auf der"
8160 LOCATE 4,6:PRINT "Sie die Daten spe
ichern wollen."
8170 LOCATE 2,8:PRINT "- Danach dr";CHR$
(220);"cken sie die Tasten"
8180 LOCATE 4,9:PRINT "'PLAY' und 'REC'
des Recorders."
8190 LOCATE 2,11:PRINT "- Dr";CHR$(220);
"cken sie 'SPACE'"
8200 IF INKEY(47)<>0 THEN 8200
8210 LOCATE 2,13:PRINT "- Die Daten werd
en gespeichert !"
8220 SPEED WRITE 1:OPENOUT "!Zehnkampfda
ten"
8230 PRINT #9,GW$
8240 PRINT #9,HGP
8250 FOR i=1 TO 10
8260 PRINT #9,WN$(i)
8270 PRINT #9,WE(i)
8280 PRINT #9,HE(i)
8290 PRINT #9,HP(i)
8300 NEXT:CLOSEOUT
8310 POKE 40300,0:da=0:RETURN
```

Referenzkarte: Calls & Pokes

Im zweiten Teil unserer Firmware Referenzkarte geht es weiter mit Einsprüngen in die Text-VDU und die Graphik-VDU.

Text-VDU

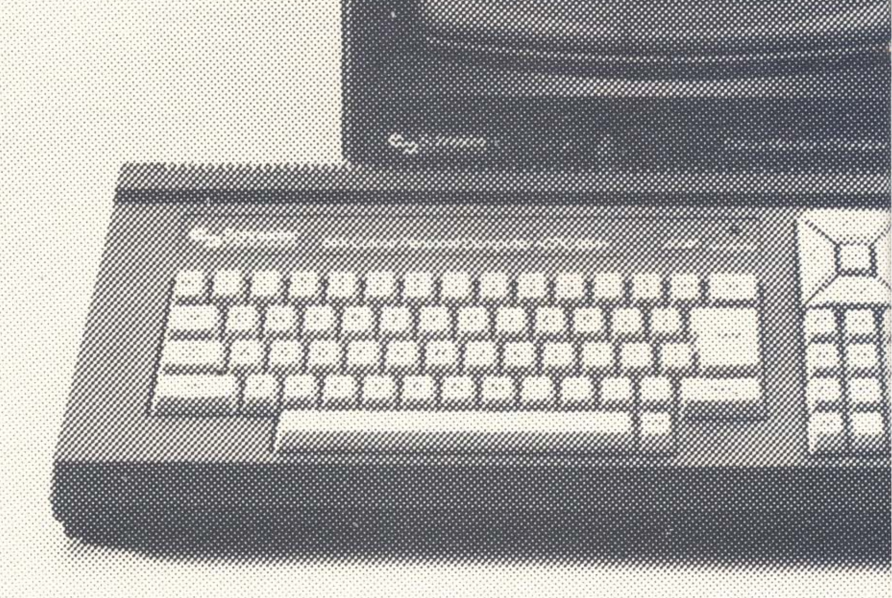
Adresse	Syntax	Auswirkung
&BB87	Einsprung: H enthält X-Koordinate. L enthält Y-Koordinate Aussprung: Kein Scroll: CARRY ON. B zerstört. Scroll hoch: CARRY OFF. B enthält &FF. Scroll tief: CARRY OFF. B enthält &00. IMMER: H= X-Koordinate. L= Y-Koordinate. Flags zerstört.	Überprüfen, ob beim Schreibversuch im Window ein Scroll vorgenommen wird.
&BB90	Einsprung: A enthält Ink Aussprung: AF. HL zerstört.	Angabe der Ink des Textstiftes
&BB93	Einsprung: Keine Bedingungen. Aussprung: A enthält Ink. Flags zerstört.	Ausgabe der Ink des Textstiftes.
&BB96	Einsprung: A enthält Farbe. Aussprung: AF. HL zerstört.	Paperfarbe setzen.
&BB99	Einsprung: Keine Bedingungen. Aussprung: A enthält Farbe. Flags zerstört.	Ausgabe der aktuellen Paperfarbe.
&BB9C	Einsprung: Keine Bedingungen. Aussprung: AF. HL zerstört.	Austausch der Ink- und Paper-Farbe (INVERSMODE).
&BB9F	Einsprung: Normal: A enthält &00. Transparent: A ist ungleich &00. Aussprung: AF. HL zerstört.	Umschaltung in Transparent-Mode.
&BBA2	Einsprung: Keine Bedingungen. Aussprung: Normal: A enthält &00. Transparent: A ist ungleich &00. Immer: DE. HL zerstört. Flags zerstört.	Ausgabe des aktuellen Cursor-Modes.
&BBB4	Einsprung: A enthält Stream-Nummer. Aussprung: A enthält vorher selektierten Stream. HL zerstört. Flags zerstört.	Bestimmen des Ein/Ausgabe-Kanals (Stream).

Graphik-VDU

Adresse	Syntax	Auswirkung
&BBBA	Einsprung: Keine Bedingungen. Aussprung: AF. BC. DE. Hl zerstört.	Vollständiges Rücksetzen der Graphik-VDU wie bei Einschaltvorgang.
&BBC0	Einsprung: DE enthält X-Koordinate. HL enthält Y-Koordinate. Aussprung: AF. BC. DE. HL zerstört.	Origin des Grafik-Cursors.
&BBC6	Einsprung: Keine Bedingungen. Aussprung: DE enthält X-Koordinate. HL enthält Y-Koordinate. AF zerstört.	Ausgabe der aktuellen Grafik-Cursor-Position.

Einblick in den CPC 664

In unserer letzten Ausgabe hatten wir bereits Teile des 664 Befehlssatzes erläutert, diese Liste werden wir nun vervollständigen. Gerade im Hinblick auf die Software-Kompatibilität sind die Anwendungen der erweiterten Basic-Befehle unerläßlich, und es stellt sich die Frage: gibt es Möglichkeiten, diese neuen 664-Befehle auf dem CPC 464 zu simulieren und so eine vollständige Kompatibilität der beiden Schneider Computer herzustellen?



Bevor wir uns jedoch dieser Frage zuwenden, erklären wir Ihnen ausführlich die noch nicht behandelten Basic-Befehle des CPC 664.

Die erweiterten Basic-Befehle und ihre Anwendung:

Graphics Paper:
definiert die Hintergrundfarbe einer graphischen Darstellung. Allerdings ist beim Zeichnen von durchgehenden Linien die Hintergrundfarbe nicht sichtbar. So lassen sich schöne Effekte erzielen, unser Beispiel wird dies verdeutlichen:

```
10 Mode 0
20 Graphics Paper 15
30 Plot 100,100:Draw 100,200:Draw
   200,200:Draw 200,100: Draw 100,
   100
40 Move 110,110: Fill 15
```

Graphics Pen:
definiert die Stiftfarbe einer graphischen Darstellung. Der Wertebereich liegt zwischen 0 und 15. Durch gleichzeitiges Setzen der Hintergrundfarbe sind ebenfalls schöne Effekte zu erzielen.

```
10 Mode 0
20 Graphics Pen 15
30 Plot 100,100: Draw 100,200: Draw
   200,200: Draw 200,100: Draw 100,
   100
40 Move 110,110: Fill 15
```

Versuchen Sie anschließend, Zeile 20 wie folgt zu ändern:

```
20 Graphics Pen 1
```

Der Unterschied wird beim Zeichnen der Umrandung deutlich, die ursprüngliche Zeichenfarbe wurde erhalten.

Mask:
Dieser Befehl formatiert das Zeichnen von Linien. Es können zwei Parameter angegeben werden, wobei der zweite nur 0 oder 1 betragen kann. Der zweite Parameter bestimmt, ob der erste zu zeichnende Punkt der Linie geplottet wird (1) oder nicht (0).

```
10 Mode 0
20 Mask 15,1
30 Graphics Pen 1
40 Plot 100,100: Draw 100,200: Draw
   200,200: Draw 200,100: Draw 100,
   100
50 Move 110,110: Fill 15
```

On Break Cont:
Dieser Befehl setzt die ESC-Taste außer Funktion. Das Programm läßt sich nach dem Starten nicht mehr unterbrechen. Es sollte also vorher abgespeichert werden. Dies ist die einfachste Form eines List- bzw. Break-Schutzes für selbstgeschriebene Basic-Programme:

```
10 On Break Cont
20 Print „Das Programm läßt sich nicht
   mehr unterbrechen"
30 For a = 1 to 5000: next: Goto 20
```

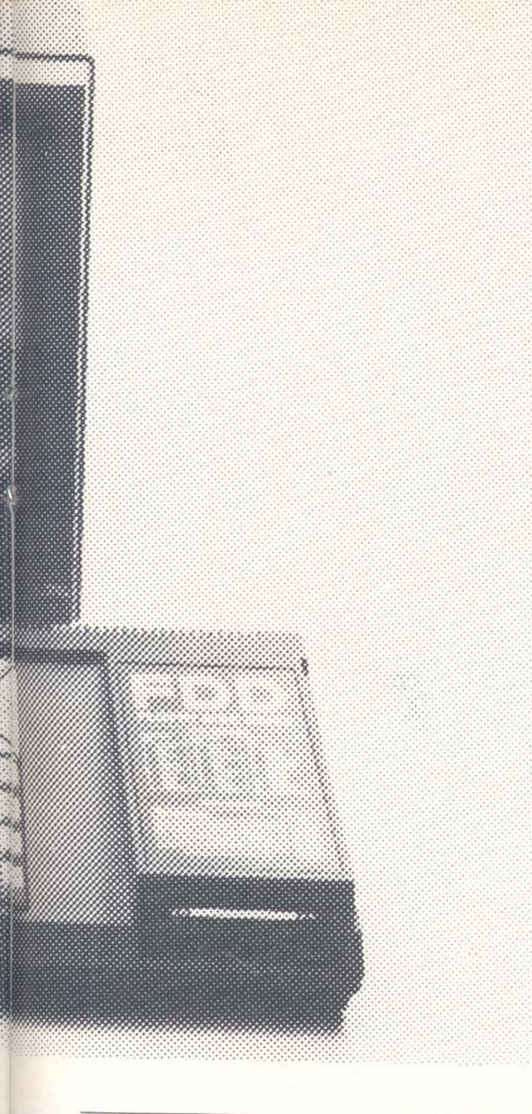
Mit einem On Break Stop-Befehl läßt sich dies innerhalb eines Programmes wieder rückgängig machen, diese Anweisung muß jedoch bereits im Programm enthalten sein. Nach dem Starten nutzt der Stop-Befehl nichts. Das waren alle erweiterten Basic-Befehle des CPC 664 und ihre Möglichkeiten der Einbindung in eigene Programme. Der neue Befehlssatz des CPC 664 unterstützt vor allem die Grafikprogrammierung. Die dafür benötigten Befehle sind wohl für Computer dieser Preisklasse einzigartig in ihrer Komplexität und Anwendungsvielfalt.

Bemerkenswertes

Inzwischen sind seit Markteinführung des CPC 664 einige Wochen vergangen und beim ständigen Arbeiten mit dem neuen CPC sind einige Dinge aufgefallen, die wir Ihnen nicht vorenthalten wollen. Zunächst erweist sich der 664 als sehr zuverlässig und vor allem bedienungsfreundlich. Gerade bei Anwendungen im professionellen Bereich (z.B. Textverarbeitung) kann man sich über die hervorragend gestylte Tastatur freuen, die wesentlich leichtgängiger ist als die 464-Tastatur. Das eingebaute 3"-Laufwerk arbeitet ebenfalls sehr zuverlässig und schnell, birgt jedoch auch nie geahnte Probleme. Da die Implementierung des Disk-Laufwerkes zwangsläufig (?) eine Verschiebung der Speicheradressen mit sich bringt, ist einige Software des CPC 464 nicht kompatibel. Wie kann hier Abhilfe geschaffen werden?

Zunächst müssen wir von der Tatsache ausgehen, daß auf dem 464 erstellte Basic-Programme grundsätzlich auf dem 664 lauffähig sind. Dennoch kann es bei einigen Programmen vorkommen, daß nach dem Einladen eines 464-Basic-Programmes in den 664 Fehlermeldungen wie „Memory full" oder „Improper argument" auftauchen.

Im Betrieb mit dem Disk-Laufwerk muß die Speicherobergrenze (HI-



MEM) heruntergesetzt werden, da sich die Speicheradressen geändert haben.

In vielen Fällen hilft schon die Anweisung HIMEM-1. Sollte trotzdem eine Fehlermeldung auftauchen, muß die Speicherobergrenze noch tiefer gesetzt werden. Hier hilft bis zum Erscheinen eines 664 Rom-Listing nur Ausprobieren. Problematischer wird es, wenn gleichzeitig Symbol-Anweisungen im Programm enthalten sind. Dann nämlich muß die Memory-Anweisung unter Berücksichtigung der Symbol-Befehle gesetzt werden – momentan gibt es hierfür noch keine Pauschal-Lösung. Zu diesem Thema erfahren Sie nächsten Monat mehr, wir arbeiten z. Z. noch an einer allgemeingültigen Lösung.

Bei Programmen, die auf bestimmte Firmware-Routinen zugreifen (z.B. bei Out-Sequenzen), ist keine Kompatibiltät möglich. Da diese, im ROM feststehenden Adressen nicht geändert werden können, hilft nur ein Umschreiben des Programms.

Was ist uns aufgefallen?

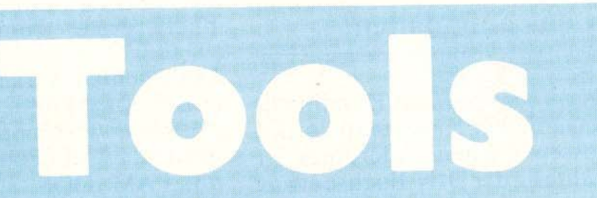
Am CPC 664 kann, so laut Handbuch, ein handelsüblicher Kassettenrekorder angeschlossen werden. Allerdings ist die Verdrahtung der DIN-Buchse etwas zu ändern, so daß die normale Verbindung zwischen Rechner und Rekorder nicht ausreicht. Die entsprechende Verdrahtung können Sie unserer Hardware-Ecke entnehmen, dort ist der Rekorderanschluß an den 664 ausführlich beschrieben.

Was bleibt, ist das Handbuch des 664. Im Gegensatz zum 464-Begleitwerk bedeutend umfangreicher. Vom Inhalt sind die wesentlichen Mängel und Fehler des 464 Handbuches ausgemerzt – Neues ist allerdings nicht hinzugekommen. Den Umfang erweitern lediglich die Disketten, Logo und CP/M-Befehle, die im einzelnen jedoch sehr knapp gehalten sind und zusätzliche Literatur erforderlich machen.

Die Software-Industrie hat sich inzwischen auf den 664 eingestellt und bietet zunehmend mehr Programme für den neuen CPC an. Nur dieser Umstand kann die peinliche Inkompatibilität von Software wettmachen und den 664 ähnlich erfolgreich werden lassen, wie den CPC 464. Im Bereich der Hardware gibt es zum Glück keine Probleme, bis auf eine Ausnahme sind alle Peripheriegeräte zum 664 komptibel.

Diskettenlaufwerke mit eigenen DOS (z.B. Vortex) können aufgrund der eingebauten 3"-Floppy nicht betrieben werden, es sei denn, als Zweitlaufwerk unter AMSDOS.

(SR)



Die Programmierung in Maschinensprache ist ein Gebiet, dem sich kein Programmierer auf Dauer entziehen kann.

MC-Programme offenbaren dem Anwender erst die volle Leistungsfähigkeit eines Rechners. Sei es als eigenständige Programme oder als schnelle Hilfsroutinen für Standard-Basic-Programme. Die Effektivität, die bei einem kurzen MC-Programm erreicht wird, läßt sich mit der von Basic nicht mehr vergleichen, da eine entsprechende Maschinenroutine die Abarbeitung bis zu 200 mal schneller durchführen kann.

Allerdings erfordert das direkte Programmieren in Maschinencode ein Genie, da hier nur mit Zahlen gearbeitet wird.

Aus diesem Grund gibt es Hilfsprogramme, mit denen sich diese Zahlenwerte in einen einigermaßen verständlichen Text übersetzen lassen. Der Assemblerprogrammierer benötigt zum Arbeiten einige Hilfsmittel, die sogenannten Tools.

Diese Tools (zu deutsch: Werkzeuge) sind Monitor, Assembler und Disassembler. Gab es am Anfang noch Schwierigkeiten ein solches Tool für den Schneider zu bekommen, so wird in der Zwischenzeit eine breite Palette an Hilfsprogrammen angeboten.
Wir werden in diesem Heft einige dieser Werkzeuge vorstellen und erläutern, für was sie überhaupt eingesetzt werden.

Monitore

MC-Monitore werden auch Debugger genannt, da sie dem Programmierer helfen sich Teile eines MC-Programmes anzusehen und eventuell Korrekturen darin vorzunehmen.
Beim Ablauf eines solchen Monitors wird dem User angegeben, in welcher Adresse welcher Wert steht. Je nach Qualität des Monitors werden zudem noch die ASCII-Werte der Adressen ausgegeben, lassen sich die Register untersuchen und können einzelne Maschinenbereiche geladen und gesaved werden.
Qualitätsmerkmale, die man beim Kauf eines Monitors beachten sollte, sind:

- Druckerausgabe
- Einzelschrittabarbeitung
- Eingabe in Hex und ASCII
- Setzen von Breakpoints
- Eingebauter Disassembler

Disassembler

Disassembler helfen dem Programmierer bei der Analyse von Programmen. Sie erzeugen aus dem Zahlenwirrwarr des Monitors einen einigermaßen lesbaren Text. Meistens sind sie in Monitore eingebunden und bilden so ein Debugger-Paket, mit dem sich der Anwender auf Fehlersuche in Programme begeben kann.

Ein Disassembler setzt die Werte der Speicherzellen in eine sogenannte Mnemonic um, das ist eine für den Anwender verständliche Interpretation des M-Codes. Dieses Assemblerlisting kann dann auf Bildschirm oder Drucker ausgegeben werden. Allerdings stimmt die Ausgabe nur in den Grundzügen mit dem Originallisting überein, da bestimmte Werte eine Doppelbedeutung haben und Texte zum Beispiel als Assemblerbefehle interpretiert werden können. Hat man die Möglichkeit, einen Disassembler direkt auf Diskette oder den Editor zu bringen, so spricht man von einem Reassembler.

Wichtige Qualitätsmerkmale, die man beim Kauf eines Disassemblers beachten sollte, sind:

- Ausgabe der Mnemonic, der ASCII- und der Speicherwerte
- Druckerausgabe
- Suchfunktionen

Assembler

Ein Assembler ist das Hauptwerkzeug, das man zum Erstellen eines MC-Programmes benötigt.
Der Anwender gibt hier, ähnlich wie bei einer Textverarbeitung, den gesamten Ablauf des MC-Programms in Z-80 Mnemonic ein.
Dabei kann er Zeilen einfügen, Adressen mit Namen vorbelegen und Kommentare zur Erläuterung des Quellcodes eingeben.
Der Quellcode ist das Listing, in dem nachher steht, was das Programm bewirkt.

Bei der eigentlichen Assemblierung analysiert der Computer den Text und setzt, wenn er eine Mnemonic findet, automatisch die entsprechende Speicherstelle auf den Wert, der dem Computer mitteilt, was er nun zu tun hat.



Je nach Qualität des Assemblers hat man einen mehr oder weniger guten Editor. Manche Assembler sind in der Lage, Texte, die mit einer Textverarbeitung geschrieben sind, zu übernehmen.

Bei einem guten Editor sind automatische Zeilennumerierung und Delete-Funktionen vorhanden.

Es gibt jedoch auch noch die Möglichkeit des Line-Assemblers. Bei dieser Art von Assembler gibt es überhaupt keinen Editor. Der eingegebene Text wird sofort nach dem Betätigen der Enter-Taste assembliert. Solche Assembler sind nur für Profis geeignet, die sich schon von vornherein darüber im klaren sind, wie ihr Listing aussieht. Meistens sind solche Assembler als Hilfsroutinen in Monitore eingebaut.

Außerdem ist noch die Anzahl der Durchläufe zu beachten, die ein Assembler macht. Man spricht von einem 1-Pass-Assembler, wenn der Quelltext ohne weitere Prüfung sofort assembliert wird. Das geht zwar schnell, hat jedoch den Nachteil, daß



Fehler mit assembliert werden.

Ein 2-Pass-Assembler prüft auch noch die syntaktische und die semantische Richtigkeit des Quelltextes und bricht bei einer Fehlermeldung den Assemblierungsvorgang ab.

Als letzter Punkt, der beachtet werden muß, sind die Assemblierungsoptionen aufzuführen. Diese Optionen werden bei der Assemblierung übergeben und teilen dem Assembler mit, was er mit dem Quelltext anzufangen hat. Möglichkeiten der Assembleroption sind z.B. Assemblierung auf Drucker, Speicher oder externen Datenträger sowie die Unterdrückung verschiedener Systemkommandos. Bei einer Assemblierung kann z.B. das Org-Kommando umgangen werden. Org legt bei den meisten Assemblern fest, in welchen Speicherbereich assembliert werden soll. Wird das Org-Kommando umgangen, so kann man den Source-Code frei in den Speicher schreiben, ohne das andere wichtige Teile des M-Codes überlappt werden.
Die wichtigen Leistungsmerkmale eines Assemblers sind also:

- leistungsfähiger Editor
- viele Kommandos zur Textbearbeitung
- anwählbare Assembleroptionen
- Eingabe in Standard Z-80-Mnemonic
- die Möglichkeit, Labels zu setzen
- Ausgabe auf Peripheriegeräte (Drukker, Floppy, Kassette).

Wie Sie sehen, gibt es beim Kauf von Tools einiges zu beachten. Natürlich ist der Bereich Werkzeuge damit noch nicht erschöpft, Zeichengeneratoren, Basic-Erweiterungen und Tastaturbelegungen zählen selbstverständlich auch zu diesen Werkzeugen, die dem Anwender die Arbeit mit dem Computer erleichtern. Wir beschränken uns in diesem Bericht jedoch nur auf die Werkzeuge, die man benötigt, um in Maschinensprache zu arbeiten.

Testbericht Star Mon

Der STAR MON ist einer der leistungsfähigsten Monitore, die es für den CPC gibt. In der Diskettenversion ist zusätzlich zum Z-80 Monitor noch ein 8080 Monitor enthalten, den man zum Debuggen von CP/M-Programmen verwenden kann.
Sowohl von der Handhabung als auch von der Leistungsfähigkeit zählt dieser Monitor zur obersten Spitzenklasse der Monitore.

Jede Eingabe wird in deutsch abgefragt, Druckerausgabe parallel zum Bildschirm, läßt sich per Tastendruck anwählen. Ein Helpmenue sowie das 30-seitige Handbuch geben schnell Aufschluß über alle Befehle, die der Anwender kennen muß. Der Monitor beinhaltet einen Debugger mit Trace-Funktion, blockweise Abarbeitung und Setzen von bis zu 3 Breakpoints. Trap-Routinen zur Ausgabe der Registerinhalte nach einem Break sind vorhanden.

Die Monitorkommandos bieten weiterhin RAM-Tests, Such- und Füll-Funktionen sowie ROM-Examine-Befehle.

Ein sehr schneller Disassembler ist eingebaut und auch ein Line-by-Line-Assembler ist vorhanden.
Besonders zu begrüßen sind bei diesem Monitor die Disk- und Kassettenoperationen, die die Startadresse eines Programms eigenständig ermitteln und abfragen, ob auch tatsächlich in diesen Bereich geladen werden soll. Das Programm ist in M-Code geschrieben, so daß auch beim Laden von Basic-Programmen keine Überlappung auftritt.
Erhältlich ist das Programm bei der Star Division auf Kassette und Disk.

Testbericht Maxam

Maxam von der Arnor Ltd. ist der Star unter den Monitoren. Eigentlich ist der Maxam ein vollständiges Entwicklungssystem. Hier sind nicht nur Monitor und Disassembler eingebaut, sondern auch noch ein vollständiger Text-Assembler mit Fullscreen-Editor.
Der Editor des Maxam ist so perfekt, das manche Textverarbeitung daneben ziemlich blaß aussieht. Nach dem Laden des Editors kehrt der Rechner in einen ganz normalen Basic-Mode zurück. Nun kann man in der Form von RSX-Erweiterungen auf alle Menues und Kommandos zugreifen. Der Monitor bietet alle Anforderungen, die man an ein solches Tool stellen kann.

Der Disassembler ist nicht nur sehr schnell, er erzeugt neben der Ausgabe der Mnemonics auch noch die ASCII-Werte und die Hex-Ausgabe einzelner Adressen.

Der Assembler erfordert keine besonderen Kenntnisse zur Handhabung. Man kann den Quelltext direkt in Basic eingeben und dann assemblieren. Ein mit Maxam erstellter Quellcode hat ungefähr folgendes Aussehen:

```
 10 REM MIT RUN 40 STARTEN
 20 REM AUSGABE EINES TEXTES
 30 CALL &5000
 40 MEMORY &5000
 50 |ASSEMBLE
 60 'LD HL, TEXT
 70 'PRINTOUT
 80 'LD A,HL    ;TEXT IN A UEBERGEBEN
 90 'CP 0       ;VERGLEICHEN DB 0
100 'RET Z      ; RETURN WENN 0
110 'CALL &BB5A;TEXT IN A AUF SCREEN
120 'INC HL     ;HL ERHOEHEN
130 'JP PRINTOUT
140 'TEXT
150 'DM "CPC INTERNATIONAL"
160 'DB 0
170 'END
```

Nach dem Start von Basic mit RUN 40 wird der Quelltext assembliert und kann dann einfach mit RUN aufgerufen werden.
Ein 50-seitiges Handbuch und Menues ermöglichen es auch hier dem Anwender, ohne Rückfragen mit dem Programmieren zu beginnen. Leider ist Maxam bisher nur in englisch erhältlich.
Ein besonderer Vorteil ist noch, daß Maxam auch auf externem ROM erhältlich ist und sich der Anwender lange Ladezeiten und Speicherverbrauch ersparen kann.
Maxam ist bei der Firma Probst als Kassetten-, Disketten- und ROM-Version erhältlich.

Testbericht CPC Bug

CPC Bug ist ein Debugger auf Kassette. Seine besonderen Vorteile liegen in den guten Trace- und Single-Step-Modis.

Der Monitor ist besonders dazu geeignet, fremde Programme zu analysieren. Dabei sind die Trace-Funktionen eine große Hilfe. Sie geben nach jedem abgearbeiteten Schritt die genauen Registerwerte an. Auch der Find-Disassembler dürfte einzigartig sein. Bei jedem gefundenen Halt oder nicht definierten Code hält er das Programm an und ermöglicht eine Modifikation, ohne daß der Rechner abstürzt. Die Einzelschritte im ROM-Bereich werden hiermit übernommen und auch eine Lokalisation eventueller Fehler ist möglich.

Die Programmierung der Monitoroptionen (Ausgabe auf Drucker etc.) werden nicht in Menues vorgenommen, sondern in sogenannten Pseudo-Registern. Diese Register sind ständig auf der Screen, so daß man sich laufend Klarheit über die momentan angewählte Option verschaffen kann. Das 40-seitige deutsche Handbuch erleichtert auch dem Anfänger die Arbeit mit diesem Monitor. Das Tool spricht deutsch und ist sowohl für Profis als auch für Anfänger gut geeignet. Erhältlich ist CPC Bug auf Kassette bei der Firma Röckrath.

Testbericht Zen

Zen ist einer der ersten Assembler, die für den CPC erhältlich waren. Trotzdem ist er nach wie vor ein Assembler-System das man zur Oberen Klasse zählen kann. Im Programm ist ein kompletter Textassembler eingebaut, der allerdings einen etwas merkwürdigen Editor hat. Von den Möglichkeiten ausgehend, die man damals hatte, arbeitet er jedoch einwandfrei und mit einer sehr guten Geschwindigkeit. Hauptnachteil des Systems sind die Hilfen, die dem Anwender gegeben werden. Das Handbuch hat zwar 50 Seiten, von diesen 50 Seiten sind jedoch 10 der Z-80-Befehlssatz und 30 das komplette Quell-Listing des Monitors. Dies ist zwar löblich für alle, die das Programm noch verbessern wollen, es bleiben jedoch nur 10 Seiten für die Anleitung übrig, und die sind auch noch in englisch. Das System ist aufgrund seiner Kürze sehr schnell geladen und der intelligente Disassembler, der die Labels gleich mitberechnet, dürfte so manchem Anwender Freude bereiten.
Das System ist Anfängern nicht zu empfehlen, aber für Fortgeschrittene jedoch recht nützlich, da der eingebaute Monitor Breakpoints verarbeiten und so optimal als Debugger eingesetzt werden kann.
Den ZEN-Assembler gibt es bei der Firma Kuma auf Kassette.

Testbericht DevPac

DevPac ist ein komplettes Assembler-Debugger-Paket von der Firma Schneider und dürfte wohl der meistverwandte Assembler auf dem Schneider sein. Seine Fähigkeiten wurden schon im letzten Heft erläutert und man kann sagen, daß er neben Maxam zur oberen Spitzenklasse zählt. Das gut geschriebene Handbuch ist auch für Anfänger zu verstehen und offensichtlich auch speziell für diese ausgelegt. Literatur über diesen Assembler zu bekommen ist aufgrund seiner Verbreitung kein Problem. Der Assembler arbeitet schnell und stellt mit dem Debugger (Monitor, Disassembler) ein hervorragendes System zur Entwicklung von Programmen in M-Code dar. Von allen getesteten Tools ist DEVPAC das einzige, dessen Location im Speicher vom Anwender frei gewählt werden kann. Der Assembler arbeitet nach dem 2-Pass-Prinzip und gewährleistet so hohe Bedienungsfreundlichkeit. Der Editor ist nicht ganz so vorteilhaft wie der des Maxam, läßt jedoch durch zahlreiche Hilfsfunktionen keine Wünsche offen. DEVPAC ist bei der Firma Schneider auf Kassette und Disk erhältlich.

Testbericht Zedis II

Zedis II ist ein Monitor in Basic. Er besitzt einige Maschinenunterprogramme, mit denen zum Beispiel die Trap-Routinen bei Breakpoints abgearbeitet werden. Das System hat jedoch als Handbuch nur ein Kassetteninlay und ist zudem noch in englisch geschrieben. Dies und die lange Ladezeit, die ein Basic-Programm von 8 Blocks Länge ausmachen, setzen den Monitor in der Wertung weit herab. Für Anfänger ist er als Spielzeug zu verwenden. Profis können mit diesem Monitor nichts anfangen.

Einziger Vorteil ist die gleichzeitige Ausgabe von Mnemonics, ASCII und Hex-Daten. Hergestellt wird der Monitor von der Firma Pride Utilities und ist auf Kassette erhältlich.

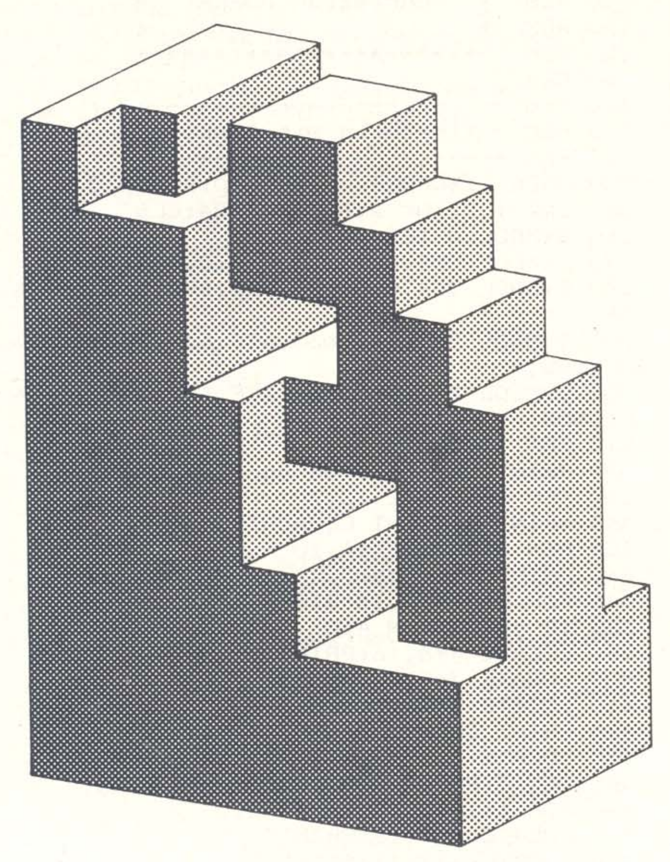
Testbericht Exmon

Der Exmon der Firma Hansesoft ist laut Beschreibung ein Hex-Monitor und Disassembler. Die Handhabung des Programms ist sehr einfach, da der Monitor nur sehr wenig Befehle kennt. Trace oder Breakpoints, die eigentlich Standard bei einem Monitor sein sollten, sind hier nicht vorhanden. Auch das Handbuch läßt mit 2 Seiten sehr zu wünschen übrig. Allerdings ist positiv anzumerken, daß diese zwei Seiten in deutsch geschrieben sind. Der Monitor dürfte kaum zu gebrauchen sein, da weder der Disassembler noch der Monitor eine Druckerausgabe haben. Das Programm ist obendrein sehr langsam und kann daher nicht empfohlen werden. Bezugsquelle: Hansesoft.

Fazit

Das Angebot an Tools ist groß, auch wenn sich die Qualitätsmerkmale von absolut „spitze" bis absolut „unbrauchbar" bewegen.
Für alle, die Assembler programmieren wollen, ist also reichhaltig Material vorhanden. Wir werden auch in weiterer Folge über neue Tools berichten und hoffen, Ihnen mit dieser kurzen Marktübersicht eine Orientierungshilfe geleistet zu haben. (TM)

Symbol-Editor



Obwohl man mit dem Schneider CPC 464 neue Zeichen ziemlich einfach selbst definieren kann, ist das Malen einer 8 x 8-Matrix sowie das Ausrechnen und Eingeben aller Zahlenwerte recht umständlich. Deshalb haben wir ein Programm geschrieben, das alle diese Arbeiten abnimmt. Und es kann noch mehr:

1. Das neu definierte Zeichen kann im Binär-, im „normalen" Dezimal- und im professionellen Hexadezimalcode dargestellt werden.

2. Gleichzeitig mit dem Eingeben in die große 8 x 8-Matrix kann das entstehende Zeichen in Originalgröße daneben abgebildet werden.

3. Man kann sich die SYMBOL-Zeile sofort auf dem Drucker ausdrucken lassen.

4. Die neu definierten Zeichen lassen sich ganz automatisch, entweder auf Kassette oder Diskette, in Form eines BASIC-Programms abspeichern, wozu man wie beim AUTO-Befehl festlegen kann, ab welcher Zeile und in welchem Abstand numeriert werden soll.

Schon der Titel wird von einem Bagger herangeschaufelt, den wir mit Hilfe des Editors definiert haben. Danach wird erfragt, ob man die SYMBOLs abspeichern will. Da sie als BASIC-Programm einschließlich eines SYMBOL-AFTER-Befehls abgespeichert werden, kann man sie auf übliche Weise laden und das Programm, in dem sie gebraucht werden, mit MERGE dazufügen oder umgekehrt.

Sie müssen noch eingeben, welche CHR$-Nummern belegt werden sollen und schon geht es los. Mit den Cursortasten wird in der 8 x 8-Matrix jedes Feld angesteuert und durch Setzen der ENTER-Taste mit Farbe gefüllt. Fehler können leicht verbessert werden, indem man zum falsch gesetzten Feld steuert und es mit DEL wieder löscht. Hat man ein SYMBOL fertig, wird die Programmzeile am Bildschirm und ggf. am Drucker ausgeprintet und es erscheint die Frage, ob man noch weitere definieren will.

Die SYMBOLs sind nach dem Programmende noch vorhanden und können mit "PRINT CHR$(Symbolnummer)" wieder aufgerufen werden. Wenn man "RUN" eingibt, werden sie jedoch gelöscht. Das Programm kann mit "GOTO 1050" wieder gestartet werden, ohne daß die vorher definierten SYMBOLs verlorengehen.

Variablenliste

j, t, i	Schleifenvariablen
zeichen$ (Spalte,Zeile):	8 x 8-Matrix des Zeichens
zeich	ASCII-Wert des zu definierenden Zeichens
s	Spalte des Cursors in der 8 x 8-Matrix
z	Zeile des Cursors in der 8 x 8-Matrix
a$	Variable für die Tastaturabfrage
a	ASCII-Wert von a$
V	Merkvariable, in welche Richtung der Cursor gesetzt wurde
x$	Bitzeile der 8 x 8-Matrix
z$ (Zeile)	endgültiger Binärwert der Zeile in der 8 x 8-Matrix
z (Zeile)	Dezimalwert von z$(Zeile)
eingabe$	wird für verschiedene Eingaben verwendet
zeile$	Programmzeile, die abgespeichert werden kann
sichern	wenn sichern=1, dann wird zeile$ abgespeichert
zeinum	Zeilennummer, mit der zeile$ abgespeichert wird
zeiab	Zeilenabstand der Programmzeilen, die abgespeichert werden
e$	Eingabevariable
name$	Name des Programms, das abgespeichert wird

Dokumentation zum Listing "Symbol-Editor"

10-160	REMs Listingkopf
200-220	REMs Titelbild
230-240	40-Zeichen-Modus und Farben werden festgelegt.
260-300	Zeichenbelegung für den Bagger
310-670	Das Titelbild wird aufgebaut. Der Bagger schiebt die Hauptüberschrift in zwei Teilen auf den Bildschirm.
1000-1030	REMs zum Programmbeginn
1040	SYMBOL AFTER löscht Bagger-SYMBOLs
1050	führt ins Abspeicher-GOSUB 4000
1060-1120	Variablen, die später wieder benötigt werden, werden vor der Definierung des nächsten Zeichens gelöscht.
1130-1140	40-Zeichen-Modus und Farben des eigentlichen Programms werden festgelegt.
1150-1200	WINDOWs 0 bis 3 und ihre Hintergrundfarben werden bestimmt.
1210-1250	Abfrage nach der zu belegenden Zeichennummer im WINDOW # 1. Der Befehl WINDOW SWAP 1 tauscht WINDOW # 0 und # 1 vorübergehend aus, muß am Ende wieder zurückgesetzt werden.
1260-1400	Das Gitternetz der vergrößerten 8 x 8-Matrix wird in WINDOW # 0 gezeichnet. Da beim Korrigieren darauf zurückgegriffen werden muß, wird es ab Zeile 1270 im GOSUB erstellt.
1410-1520	Im WINDOW # 3 und # 1 wird die Anleitung zum Setzen und Löschen eines Punktes sowie zur Cursor-Steuerung gegeben.
1530	Die Variablen s und z werden gleich 1 gesetzt (für späteren LOCATE-Befehl notwendig).
1540	Der Cursor-Punkt erscheint links oben in der 8 x 8-Matrix (WINDOW # 0) in Form eines gelben Karofeldes.
2000-2020	REMs zur Tastaturabfrage
2030	Warten auf Tastendruck

2040 a erfaßt den ASCII-Wert der gedrückten Taste.

2050-2080 Je nach gedrückter Cursor-Taste verändern sich die LOCATE-Variablen s und z; die Variable v wird festgelegt.

2090 Wenn eine Cursor-Taste gedrückt, also v einen Wert angenommen hat, wird im GOSUB 2140-2350 der Cursor-Punkt bewegt.

2100 Das Drücken der "eckigen Klammer zu" beendet die Belegung des betreffenden SYMBOLs und führt zu weiteren Abfragen in Zeile 3420.

2110 Mit Drücken der ENTER-Taste wird im GOSUB 3170-3240 ein Punkt gesetzt.

2120 Mit der DELETE-Taste kann im GOSUB 3250-3320 ein gesetzter Punkt wieder gelöscht werden.

2130 Das Drücken einer falschen Taste führt an den Anfang der Tastaturabfrage zurück.

2140-2160 REMs zu den GOSUBs Cursorsteuerung

2170 Zunächst wird durch GOSUB 3330-3410 eine Randüberschreitung des Cursor-Punktes verhindert.

2190-2340 In Abhängigkeiten von v (Cursortasten-Variable) wird der Cursor-Punkt bewegt.

3000-3030 REMs zum Ausdruck des Zeichens in Originalgröße

3040-3150 Das Zeichen, das auf der groß dargestellten Matrix definiert wird, entsteht gleichzeitig daneben in Normalgröße.

3160 Rücksprung zur Tastaturabfrage

3170-3240 GOSUB zum Setzen eines Punktes

3250-3320 GOSUB zum Löschen eines gesetzten Punktes

3340-3410 GOSUB zur Vermeidung einer Randüberschreitung

3420-3440 REMs zur Abfrage nach dem gewünschten Zahlencode

3450-3520 Abfrage, ob das Zeichen binär, dezimal oder hexadezimal festgelegt werden soll.

3530-3550 Der Rechner wartet auf eine Eingabe, die durch UPPER$ in Großbuchstaben umgewandelt wird.

3560-3640 Je nach der gewünschten Art der Belegung wird diese auf dem Bildschirm gezeigt.

3650 Wenn die Definierung abgespeichert werden soll, springt der Computer in das GOSUB 4250-4260.

3660-3690 Falls ein Ausdruck auf dem Drucker verlangt wird, geschieht dies durch den Befehl PRINT # 8.

3700-3730 Abfrage nach weiteren zu belegenden Zeichen.

3740 Programmende mit Schließen der Abspeicher-Datei.

4000-4020 REMs zum GOSUB Abspeichern.

4030-4200 Abfragen, die für das Abspeichern nötig sind. Die Abspeicherung erfolgt automatisch in Form eines BASIC-Programms.

4210-4240 Das Sichern der Zeichenbelegung wird eröffnet (OPEN-OUT-Befehl in 4210) und beim Programmende in Zeile 3740 wieder geschlossen. Der SYMBOL-AFTER-Befehl (Zeile 4220) muß dabei mit abgespeichert werden.

4250-4260 GOSUB zum Abspeichern eines einzelnen Zeichens als Programmzeile.

```
10 REM  ************************
20 REM  *                      *
30 REM  *    SYMBOL-EDITOR     *
40 REM  *                      *
50 REM  ************************
60 REM  *                      *
70 REM  *  Schneider CPC 464   *
80 REM  *                      *
90 REM  ************************
100 REM  *                      *
110 REM  *     (c) 1985 by      *
120 REM  *     Joachim und      *
130 REM  *   Christian HOFER    *
140 REM  *                      *
150 REM  ************************
160 REM
200 REM -------------------------
210 REM - Titelbild mit Symbols -
220 REM -------------------------
230 MODE 1:BORDER 0:INK 0,0:PAPER 0
240 INK 1,2:INK 2,6:INK 3,24:CLS
260 SYMBOL AFTER 252
270 SYMBOL 255,0,0,3,0,0,0,0,&41
280 SYMBOL 254,0,0,&FF,&82,&9A,&9A,&8A,&
DE
290 SYMBOL 253,&61,&33,&3F,&3F,&3A,&67,&
CF,&85
300 SYMBOL 252,&FF,&83,&FF,&FF,&AA,&FF,&
FE,&55
310 PEN 1:LOCATE 9,4
320 PRINT "Definieren von Zeichen"
330 PEN 2:LOCATE 13,7
340 PRINT "LEICHT GEMACHT"
350 PEN 3:LOCATE 16,10
360 PRINT "mit dem"
370 FOR j=32 TO 13 STEP -1
380 IF j=32 THEN 410
390 LOCATE j+8,16:PRINT" "
400 LOCATE j+8,17:PRINT" "
410 LOCATE j+6,16:PEN 3
420 PRINT CHR$(255);CHR$(254)
430 LOCATE j,17:PEN 2:PRINT"Symbol";
440 PEN 3:PRINT CHR$(253);CHR$(252)
450 FOR t=1 TO 125:NEXT t,j
460 LOCATE j+7,16:PRINT"  "
470 LOCATE j+7,17:PRINT"  "
480 FOR j=31 TO 19 STEP -1
490 IF j=31 THEN GOTO 520
500 LOCATE j+9,16:PRINT" "
510 LOCATE j+9,17:PRINT" "
520 LOCATE j+7,16:PEN 3
530 PRINT CHR$(255);CHR$(254)
540 LOCATE j,17:PEN 2
550 PRINT "-Editor";
560 PEN 3:PRINT CHR$(253);CHR$(252)
570 FOR t=1 TO 125:NEXT t,j
580 PEN 1:LOCATE 3,24
590 PRINT "(c) by Joachim and Christian
Hofer"
600 FOR i=15 TO 19
610 PEN 2:LOCATE 11,i
620 PRINT STRING$(17,143)
630 NEXT i
640 PEN 3:LOCATE 13,17
650 PRINT CHR$(22)CHR$(1);
660 PRINT"Symbol-Editor";CHR$(22);CHR$(0
)
670 FOR i=1 TO 3000:NEXT
680 REM
1000 REM =====================
1010 REM === Programmbeginn ===
1020 REM =====================
1030 REM
1040 SYMBOL AFTER 32
1050 GOSUB 4000
1060 FOR i=1 TO 8
1070 FOR j=1 TO 8
1080 zeichen$(i,j)=""
1090 NEXT j,i
1100 FOR i=1 TO 8
1110 z(i)=0:z$(i)=""
1120 NEXT i
1130 INK 0,1:INK 1,20:INK 2,0:INK 3,24
1140 MODE 1:BORDER 1:PAPER 2:PEN 1
1150 WINDOW 1,9,1,9
1160 WINDOW #1,1,40,9,25
```

```
1170 WINDOW #2,9,12,1,8
1180 WINDOW #3,13,40,1,8
1190 PAPER #1,0:PAPER #2,0:PAPER #3,3
1200 CLS:CLS #1:CLS #2:CLS #3
1210 WINDOW SWAP 1
1220 PRINT:PRINT"Das zu definierende Zei
chen soll auf"
1230 PRINT"CHR$(    ) gelegt werden."
1240 LOCATE 6,3:INPUT "",zeich
1250 WINDOW SWAP 1
1260 GOSUB 1270:GOTO 1410
1270 ORIGIN 0,400-8*16
1280 DRAW 8*16,0
1290 DRAWR 0,8*16-1
1300 DRAWR -8*16,0
1310 DRAWR 0,-8*16
1320 FOR i=1 TO 8
1330 MOVE i*16,0
1340 DRAWR 0,8*16
1350 NEXT
1360 FOR i=1 TO 8
1370 MOVE 0,i*16
1380 DRAWR 8*16,0
1390 NEXT
1400 RETURN
1410 WINDOW SWAP 3:PEN 0
1420 PRINT:PRINT"<ENTER>:Setzen eines Pu
nktes";
1430 PRINT"<DEL>:Loeschen eines Punktes"
;
1440 PRINT"  ";CHR$(240);"  :Cursor nach
 oben"
1450 PRINT"  ";CHR$(242);"  :Cursor nach
 links"
1460 PRINT"  ";CHR$(241);"  :Cursor nach
 unten"
1470 PRINT"  ";CHR$(243);"  :Cursor nach
 rechts"
1480 WINDOW SWAP 3:WINDOW SWAP 1:PEN 3
1490 LOCATE 1,9
1500 PRINT"Wenn das Zeichen fertig ist,
dann"
1510 PRINT"druecken Sie ";CHR$(24);" ";C
HR$(93);" ";CHR$(24)
1520 WINDOW SWAP 1
1530 s=1:z=1
1540 PEN 3:PRINT CHR$(207)
2000 REM ----------------------
2010 REM --- Tastaturabfrage ---
2020 REM ----------------------
2030 a$=INKEY$:IF a$="" THEN 2030
2040 a=ASC(a$)
2050 IF a=240 THEN z=z-1:v=1
2060 IF a=241 THEN z=z+1:v=2
2070 IF a=242 THEN s=s-1:v=3
2080 IF a=243 THEN s=s+1:v=4
2090 IF v<>0 THEN GOSUB 2140:GOTO 2000
2100 IF a=93 THEN 3420
2110 IF a=13 THEN GOSUB 3170:GOTO 3000
2120 IF a=127 THEN GOSUB 3250:GOTO 3000
2130 GOTO 2000
2140 REM
2150 REM Gosubs Cursorsteuerung
2160 REM ---------------------
2170 GOSUB 3330:PEN 3
2180 LOCATE s,z:PRINT CHR$(207):PEN 1
2190 IF v=1 THEN GOSUB 2240
2200 IF v=2 THEN GOSUB 2270
2210 IF v=3 THEN GOSUB 2300
2220 IF v=4 THEN GOSUB 2330
2230 v=0:GOSUB 1270:RETURN
2240 IF zeichen$(s,z+1)="" OR zeichen$(s
,z+1)="0" THEN LOCATE s,z+1:PRINT" "
2250 IF zeichen$(s,z+1)="1" THEN LOCATE
s,z+1:PRINT CHR$(143)
2260 RETURN
2270 IF zeichen$(s,z-1)="" OR zeichen$(s
,z-1)="0" THEN LOCATE s,z-1:PRINT" "
```

Sie wollen uns ein Programm zusenden? Folgende Regeln sollten Sie beachten:

1) Schicken Sie Ihre Programme auf Kassette oder Diskette. Dabei mehrmals hintereinander abspeichern.

2) Legen Sie eine ausführliche Programmbeschreibung bei, aus der eindeutig der Sinn und Zweck des Programmes hervorgeht. Eine Variablenliste ist sehr wichtig, in ihr sollten alle im Programm definierten Variablen mit ihrer Verwendung aufgeführt sein.
Beispiel: n$ = Nachname
g = glücklicher usw.

3) Wenn Sie gleichzeitig Besitzer eines Druckers sind, schicken Sie auf jeden Fall ein Programmlisting mit.

4) Vergessen Sie Ihren Absender nicht! Vollständige Anschrift, wenn möglich auch die Telefonnummer, ersparen unnötige Verzögerungen bei eventuellen Nachfragen unsererseits.

5) Behalten Sie auf jeden Fall eine Kopie Ihres Programmes, unvorhersehbare Umstände, wie z.B. Bandriß der Kassette, werden somit nicht zur Katastrophe.

6) Wenn Sie sogar mehrere Programme geschrieben haben und diese veröffentlichen wollen, nehmen Sie bitte für jedes Programm eine separate Kassette (mehrmals abspeichern!). Damit erleichtern Sie unsere Arbeit wesentlich und eine schnellere Bearbeitung wird möglich.

7) Versehen Sie das Programm mit den sehr wichtigen REM-Statements. Das fördert nicht nur die Übersicht für uns und unsere Leser, sondern unterstützt zusätzlich das strukturierte Programmieren. Merke: Auch erfahrene Programmierer verwenden REM-Zeilen, das Programm sollte schließlich jeder verstehen und anwenden können.

8) Vermeiden Sie möglichst Variablennamen, wo leicht l und i oder 0 und o verwechselt werden können. Variablen sollten grundsätzlich in Kleinbuchstaben geschrieben werden und möglichst verständlich sein. Die Variable butter sagt mehr aus, als nur bu.

9) Bieten Sie das Programm niemals verschiedenen Verlagen gleichzeitig an! Mit der Veröffentlichung und dem daraus resultierenden Honorar, gehen die Urheberrechte an den Verlag über! Sollte Ihr Programm dann noch in einer anderen Zeitschrift abgedruckt werden, liegt ein Verstoß gegen das Wettbewerbsrecht vor.

10) Jedes bei uns ordnungsgemäß eingegangene Programm wird ausführlich begutachtet und getestet. Sie erhalten von uns dann umgehend Bescheid, ob Ihr Programm veröffentlicht wird oder nicht. Wenn Sie das Programm nach unserer Begutachtung zurückhaben wollen, legen Sie bitte der Einsendung einen frankierten und adressierten Rückumschlag bei. Die Kosten einer eventuellen Rücksendung sind in jedem Fall vom Einsender zu tragen.

Falls wir uns für den Abdruck Ihres Programmes entschieden haben, setzen wir uns unverzüglich mit Ihnen in Verbindung und besprechen Ihr verdientes Honorar!
Übrigens wäre es nett, wenn Sie uns auch ein paar persönliche Daten schreiben würden (z.B. Ihr Alter, Tätigkeit, wie und wann kam die Idee zu dem Programm, was alles machen Sie mit dem CPC usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre „Schneider CPC international" auf Ihre Beiträge.

Einsendungen bitte an: Data Media GmbH -Bereich Verlag-, Fuldaer Straße 6, 3440 Eschwege

```
2280 IF zeichen$(s,z-1)="1" THEN LOCATE
s,z-1:PRINT CHR$(143)
2290 RETURN
2300 IF zeichen$(s+1,z)="" OR zeichen$(s
+1,z)="0" THEN LOCATE s+1,z:PRINT" "
2310 IF zeichen$(s+1,z)="1" THEN LOCATE
s+1,z:PRINT CHR$(143)
2320 RETURN
2330 IF zeichen$(s-1,z)="" OR zeichen$(s
-1,z)="0" THEN LOCATE s-1,z:PRINT" "
2340 IF zeichen$(s-1,z)="1" THEN LOCATE
s-1,z:PRINT CHR$(143)
2350 RETURN
3000 REM ----------------------
3010 REM -- Belegen & Printen --
3020 REM ----  des Zeichens ----
3030 REM ----------------------
3040 FOR j=1 TO 8
3050 x$=""
3060 FOR i=1 TO 8
3070 IF zeichen$(i,j)="" THEN zeichen$(i
,j)="0"
3080 x$=x$+zeichen$(i,j)
3090 NEXT i
3100 z$(j)="&x"+x$
3110 z(j)=VAL(z$(j))
3120 NEXT j
3130 SYMBOL zeich,z(1),z(2),z(3),z(4),z(
5),z(6),z(7),z(8)
3140 LOCATE #2,3,3
3150 PRINT #2,CHR$(zeich)
3160 GOTO 2000
3170 REM
3180 REM Gosub Bit setzen
3190 REM ----------------
3200 GOSUB 3330
3210 LOCATE s,z:PEN 1
3220 PRINT CHR$(143)
3230 zeichen$(s,z)="1"
3240 RETURN
3250 REM
3260 REM Gosub Bit loeschen
3270 REM ------------------
3280 GOSUB 3330
3290 LOCATE s,z:PRINT " "
3300 zeichen$(s,z)="0"
3310 GOSUB 1270
3320 RETURN
3330 REM
3340 REM Gosub 8*8-Matrix
3350 REM    einhalten
3360 REM ----------------
3370 IF s<1 THEN s=1
3380 IF s>8 THEN s=8
3390 IF z<1 THEN z=1
3400 IF z>8 THEN z=8
3410 RETURN
3420 REM
3430 REM Abfrage Zahlensystem
3440 REM --------------------
3450 WINDOW SWAP 1:CLS:PEN 1
3460 PRINT:PRINT "Soll das Zeichen ";
3470 PEN 3:PRINT "d";
3480 PEN 1:PRINT "ezimal, ";
3490 PEN 3:PRINT "h";
3500 PEN 1:PRINT "exadezimal":PRINT "ode
r ";
3510 PEN 3:PRINT "b";
3520 PEN 1:PRINT "inaer definiert werden
?"
3530 eingabe$=""
3540 WHILE eingabe$="":eingabe$=INKEY$:W
END
3550 eingabe$=UPPER$(eingabe$)
3560 IF LEFT$(eingabe$,1)<>"D" THEN GOTO
 3580
3570 zeile$="SYMBOL"+STR$(zeich)+","+STR
$(VAL(z$(1)))+","+STR$(VAL(z$(2)))+","+S
TR$(VAL(z$(3)))+","+STR$(VAL(z$(4)))+","
+STR$(VAL(z$(5)))+","+STR$(VAL(z$(6)))+"
,"+STR$(VAL(z$(7)))+","+STR$(VAL(z$(8)))
3580 IF LEFT$(eingabe$,1)<>"B" THEN GOTO
 3600
3590 zeile$="SYMBOL"+STR$(zeich)+",&x"+B
IN$(VAL(z$(1)))+",&x"+BIN$(VAL(z$(2)))+"
,&x"+BIN$(VAL(z$(3)))+",&x"+BIN$(VAL(z$(
4)))+",&x"+BIN$(VAL(z$(5)))+",&x"+BIN$(V
AL(z$(6)))+",&x"+BIN$(VAL(z$(7)))+",&x"+
BIN$(VAL(z$(8)))
3600 IF LEFT$(eingabe$,1)<>"H" THEN GOTO
 3630
3610 zeile$="SYMBOL"+STR$(zeich)+",&"+HE
X$(VAL(z$(1)))+",&"+HEX$(VAL(z$(2)))+",&
"+HEX$(VAL(z$(3)))+",&"+HEX$(VAL(z$(4)))
+",&"+HEX$(VAL(z$(5)))+",&"+HEX$(VAL(z$(
6)))+",&"+HEX$(VAL(z$(7)))+",&"+HEX$(VAL
(z$(8)))
3620 IF LEFT$(eingabe$,1)<>"D" AND LEFT$
(eingabe$,1)<>"B" AND LEFT$(eingabe$,1)<
>"H" THEN GOTO 3530
3630 PRINT:PRINT"So wird das Zeichen def
iniert:"
3640 PRINT zeile$
3650 IF sichern=1 THEN GOSUB 4250
3660 PRINT:INPUT"Wird ein Ausdruck auf D
rucker verlangt";eingabe$
3670 eingabe$=UPPER$(eingabe$)
3680 IF LEFT$(eingabe$,1)<>"J" THEN 3700
3690 PRINT#8,zeile$
3700 CLS:PRINT
3710 INPUT"Noch ein Zeichen zu definiere
n";eingabe$
3720 eingabe$=UPPER$(eingabe$)
3730 IF LEFT$(eingabe$,1)="J" THEN GOTO
1060
3740 CLOSEOUT:MODE 1:END
4000 REM -----------------------
4010 REM -- Gosub Abspeichern --
4020 REM -----------------------
4030 WINDOW SWAP 1
4040 BORDER 1:INK 0,1:PAPER 0:INK 1,20:P
EN 1:CLS
4050 INPUT "Sollen die Zeichen abgespeic
hert werden";eingabe$
4060 eingabe$=UPPER$(eingabe$)
4070 eingabe$=LEFT$(eingabe$,1)
4080 IF eingabe$="J" THEN sichern=1 ELSE
 4240
4090 PRINT:INPUT "Ab welcher Zeilennumme
r";zeinum
4100 PRINT:INPUT "Mit welchem Zeilenabst
and";zeiab
4110 PRINT:PRINT "Auf Kassette (druecken
 Sie K) oder"
4120 PRINT "Diskette (druecken Sie D)"
4130 e$="":WHILE e$="":e$=INKEY$:WEND
4140 IF UPPER$(e$)="K" THEN GOTO 4190
4150 IF UPPER$(e$)<>"D" THEN GOTO 4130
4160 PRINT:INPUT "Mit welchem Namen";nam
e$
4170 name$=LEFT$(name$,8)+".bas"
4180 GOTO 4210
4190 INPUT "Mit welchem Namen";name$
4200 name$=LEFT$(name$,16)
4210 OPENOUT name$
4220 zeile$="SYMBOL AFTER 32"
4230 GOSUB 4250
4240 CLS:WINDOW SWAP 1:RETURN
4250 PRINT#9,STR$(zeinum)+zeile$
4260 zeinum=zeinum+zeiab:RETURN
```



TV-Anschluß für alle Schneider

Viele Schneider Besitzer haben sich einen CPC mit Grünmonitor zugelegt. Für Anwendungsprogramme ist diese Konfiguration sicherlich auch gut geeignet, bei Spielen jedoch ist ein farbiges Bild oft wünschenswert.

Alternativ zu diesem Problem hat die Firma Schneider das separate Netzteil MP-1 herausgebracht, in dem ein UHF-Modulator untergebracht ist.
Allerdings wird sich so mancher Anwender überlegen, den Preis für ein solches Netzteil zu entrichten, wenn er höchstens einmal im Monat mit seinem Computer spielen kann.

Haben Sie jedoch einen Fernseher der einen Scart-Eingang hat, so können Sie sich im Selbstbau eine wesentlich billigere Lösung schaffen. Voraussetzung dazu ist jedoch, daß der Grünmonitor mitläuft, da sonst keine Versorgungsspannung anliegt.
Die Scart-Buchse ist eine sehr junge Steckerentwicklung. Sie macht jedoch große Fortschritte, ein zumindest europäischer Standard zu werden.

Scart ist die Abkürzung für **S**yndicat des **C**onstructeurs d'**A**pareils **R**adio recepteurs et **T**eleviseurs, auch Euro-AV-Verbindung genannt. Sie ist für alle im Augenblick denkbaren analogen Video- und Audiosignale vorbereitet; auch für RGB. Zwei Datenleitungen für digitalisierte Signale stehen zur Verfügung. Eine genaue Pinbelegung entnehmen Sie der Zeichnung.

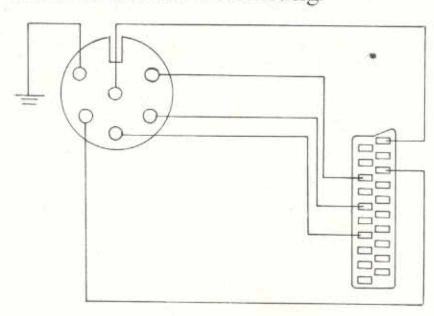


Diagramm 2 zeigt, wie der Stecker mit Ihrem Rechner verbunden werden muß. Der Übertrag der Datenleitungen ist wie folgt:

Pin-Nummer	CPC	Pin-Nummer	Scart
1	rot	15	rot
2	grün	11	grün
3	blau	7	blau
4	sync.	16	Austas.
5	Masse	21	Abschi.
6	Leuchtdichte	20	Video

Um die Schaltung in Betrieb zu nehmen, verbinden Sie Ihren Schneider mit dem Monitor. Danach entfernen Sie den 6-poligen DIN-Stecker aus dem Rechner und verbinden ihn mit dem neuen Adapterkabel. Stecken Sie den Scart-Stecker in die entsprechende Buchse Ihres Fernsehers und betätigen Sie die AV-Taste. (TM)

Eprom-Programmierer

- „schießen" Sie Ihr eigenes Betriebssystem!



Wer sich die technische Beschreibung seines Schneider Computer mal etwas genauer durchgelesen hat, ist bestimmt auf den Hinweis gestoßen, daß für die CPC-Rechner sogenannte Steckmodule vorgesehen sind. Ein Steckmodul beinhaltet die Software, die normalerweise von einem Datenträger (Kassette, Diskette) in den Computer eingeladen wird.

In einem Steckmodul befindet sich meist ein ROM (nur Lesespeicher) oder ein EPROM. Ein EPROM ist ein elektrisch löschbarer und programmierbarer Festwertspeicher, der mit Hilfe eines EPROM-Programmiergerätes programmiert werden kann.

Die Firma Dobbertin hat nun mit dem System 4003 einen universellen EPROM-Programmer für den CPC 464/664 entwickelt, der alle gängigen EPROM-Typen programmiert. Nachfolgend ein Auszug der möglichen Typen:
2716, 2732, 2732A, 2764, 27128, 27256, 2508, 2516, 2532, 2564.

Die Hardware

Um ein EPROM-Programmiergerät am CPC zu betreiben, benötigt man zuerst eine serielle Schnittstelle. Diese Schnittstelle wird in Form einer Interface-Karte mitgeliefert, die einfach in den Floppy-Anschluß des CPC eingesteckt wird. Das EPROM-Programmiergerät wird über ein mitgeliefertes, ca. 60 cm langes Flachbandkabel mit der Interface-Karte verbunden. Trotzdem ist der gleichzeitige Betrieb der Diskettenstation möglich, da die Steckerleiste an der Rückseite ausgeführt ist (diese Aufmerksamkeit ist man nicht von allen Herstellern gewohnt!).

Zur Aufnahme der verschiedenen EPROM-Typen ist an der Oberseite des Gerätes ein 28poliger Textool-Sockel vorhanden. Die seriell übergebenen Daten werden in Schieberegister übernommen, wo sie unter Kontrolle der mitgelieferten, deutschsprachigen Software an den Textool-Sockel durchgeschaltet werden. Dadurch wird die Anpassung einiger „exotischer" EPROM-Typen wesentlich vereinfacht. Allerdings benötigt man schon reichlich Erfahrung in der Assembler-Programmierung, um eine derartige Anpassung vorzunehmen. Für diesen Fall bietet der Hersteller ein dokumentiertes Listing an, das gegen einen geringen Unkostenbeitrag erhältlich ist.

Die für die einzelnen EPROMs benötigten, unterschiedlichen Programmierspannungen werden im Gerät erzeugt, die Typ-Auswahl eines EPROMs wird softwaremäßig vorgenommen. Über die zwei LED-Anzeigen kann zu jeder Zeit der Betriebszustand des Programmiergerätes abgelesen werden.

Die Software

Die Software wird auf Kassette mitgeliefert und bietet alle benötigten Optionen einer EPROM-Programmierung.

```
>> Universeller EPROM-Programmer 4003<<

 < T >        Typ angeben
 < R >        Einlesen von Eprom
 < P >        Programmieren
 < L >        Lesen von Cassette
 < S >        Schreiben auf Cassette
 < D >        Dumpen von Eprom
 < M >        Dumpen von RAM
 < V >        Vergleichen
 < C >        Löschtest
 < E >        Ende

 Bitte wählen Sie aus -->
```

Nachfolgend eine kurze Beschreibung der einzelnen Funktionen:

Funktionen:

T - Bei der Auswahl eines Typs werden automatisch sämtliche „Default"-Werte (z.B. max. EPROM-Adresse, Programmierspannung...) in die entsprechenden Variablen übernommen.

R - Der Inhalt des EPROMs wird in den Hauptspeicher des CPC eingelesen.

P - Der Inhalt des RAM wird ab der angegebenen Startadresse in das EPROM programmiert.

L - Auf Kassette abgelegte Binärfiles werden in den Speicher geladen.

S - Speicherinhalte werden als Binärfiles auf Cassette geschrieben.

D - Der Inhalt eines EPROMs wird als HEX-Dump auf dem Bildschirm ausgegeben.

M - Der Inhalt des RAM wird als HEX-Dump auf dem Bildschirm ausgegeben.

V - Die Inhalte von RAM und EPROM werden miteinander verglichen.

C - Der Inhalt des EPROMs wird auf $FF überprüft.

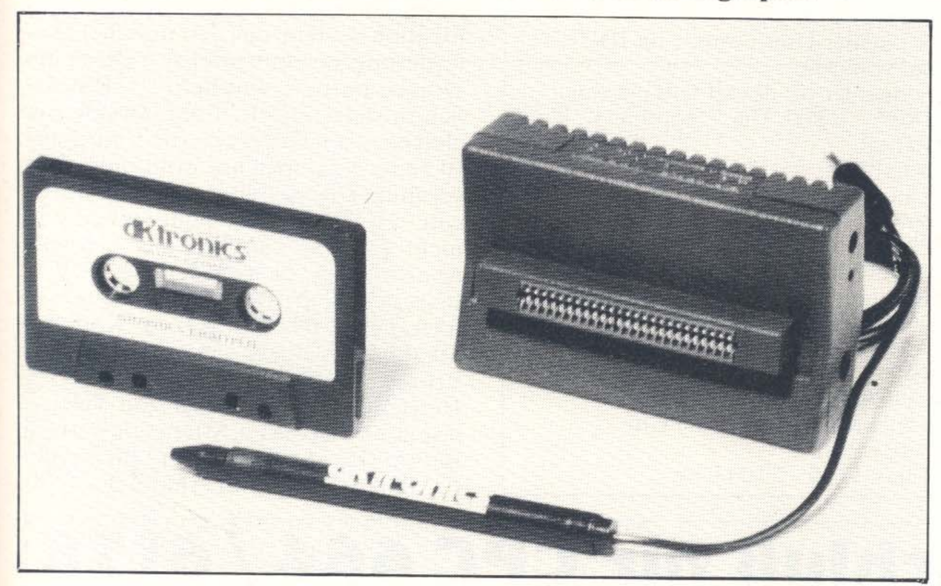
E - Das Programm wird abgebrochen.

Fazit:
Das EPROM-Programmiergerät 4003 erwies sich in der Testphase als sehr zuverlässig. Voraussetzung für eine effektive Programmierung von EPROMs ist allerdings die Assemblersprache, die man zumindest beherrschen sollte. Auf diesem Wege ist es dann auch möglich, eigene Software auf EPROM zu „schießen" und so per einfache Umschaltung zwischen mehreren ROMs hin- und herzuschalten, ohne daß unnötige Ladezeiten von Datenträgern in Kauf genommen werden müssen. Eine sicherlich interessante Anwendung ist das Erstellen eines eigenen Betriebssystem, so wäre z.B. der Mangel mit der relativen Dateiverwaltung beim Disk-Betrieb im Eigenbau zu beheben. Das EPROM-Programmiergerät 4003 gibt es als Fertiggerät zum Preis von ca. DM 290,-. Als Bausatz ist das Gerät zum Preis von ca. DM 240,- erhältlich. (SR)

Supergrafik

Malen wie Leonardo da Vinci

Daß der Schneider CPC ein grafisches Talent ist, braucht aufgrund seiner reichhaltigen Befehle zur Grafikerzeugung nicht extra erwähnt werden. Wie man die Handhabung dieser grafischen Eigenschaften jedoch ohne große Vorkenntnisse bis zur Perfektion nutzen kann, das zeigt erst der dk'tronics Lightpen.



Zur Hardware:

Zum Lieferumfang des Systems gehören die Kassetten-Software, eine Expansion Box mit durchgeführtem Systembus, sowie der Lightpen. Zum Anschluß steckt man die Box an den Expansion Port des CPC 464. Besitzer des Diskettenlaufwerkes können ihre Peripherie an den durchgeführten Bus anschließen. Nachdem der Lightpen über einen Klinkenstecker mit der Expansion Box verbunden wurde, ist zunächst die Software zu laden. Diese Software ist eine der schnellsten und anwendungsfreundlichsten Grafikprogramme, die bisher auf dem CPC-Prüfstand vorlagen. Circle, Rectangle und Fill-Befehle arbeiten mit hoher Präzision und unglaublichen Geschwindigkeiten. Die Steuerung des Programms erfolgt über Pictogramm-unterstützte Menues, die den Anwender durch das komplette Programm führen. Insgesamt gibt es sechs dieser Menues. Funktionen werden über den Lightpen angewählt und durch Drücken der Enter-Taste bestätigt.

So können z.B. Punkte gesetzt werden, Farbe angewählt, die Strichstärke bestimmt oder Lade- und Save-Kommandos ausgeführt werden. Einzigartig dürfte die Möglichkeit sein, Bildschirmausschnitte frei zu verschieben, vergrößern, verkleinern oder im Speicher auf einem „Notizblock" abzulegen, um bei späteren Bedarf problemlos wieder darauf zurückgreifen zu können.

Weiterhin ist noch anzumerken, daß sich mit der Option „Rubber Banding" auch Halb- und Viertelkreise erzeugen lassen. Bilder können auch ohne Lightpen wieder eingeladen werden. Durch die Möglichkeiten der Texteingabe, die es erlaubt, Schriftzüge horizontal und vertikal auf die Screen zu schreiben, lassen sich so schöne Titelbilder für eigene Programme erstellen. Damit sind die Möglichkeiten des Lightpen aber noch lange nicht erschöpft. Durch wenige Basic-Befehle läßt sich die Position des Lightpen auch ohne System-Software abfragen, dies ist z.B. für einen Programmierer wichtig, der sich für seine Anwendersoftware einfache Menuesteuerungen schreiben will. Anzumerken wäre noch, daß die Software sowohl mit Floppy als auch Kassette läuft und daß die Kompatibilität zum Grünmonitor und MP-1 besteht.

Fazit:

Für ca. DM 99,– erhält man ein leistungsfähiges Grafikpaket, das keine Wünsche offen läßt. Die Bedienung des Systems ist denkbar einfach und schon nach kurzer Zeit ist jeder in der Lage, ansehnliche Grafiken zu erzeugen und zu verwalten. In dem ca. 30-seitigen Bedienungsheft sind u.a. Anleitungen für Hardcopy, Einbinden in eigene Programme und Hinweise zur effektvollen Grafikerzeugung enthalten. Der dk'tronics Lightpen ist im Vertrieb der Data Media und kann als sehr empfehlenswert bezeichnet werden. (SR/TM)

Stichwort Mailbox: Teleterminal 300s

Die Möglichkeit der Datenfernübertragung mit dem Schneider CPC eröffnet vielen Anwendern neue Kommunikationswege. Der Datentransfer per Telefon erfreut sich allgemein großer Beliebtheit. Dieses Medium ist längst nicht nur den „Hackern" vorbehalten, wie die Benutzer dieser Datensysteme oftmals tituliert werden.

Zumindest in der Bundesrepublik scheint sich ein Trend durchzusetzen, der dieses neue Medium in einer Weise nutzt, die mit dem puren „Herumstöbern" in fremden Datenbanken überhaupt nichts mehr zu tun hat. Im Gegenteil, immer mehr Anwender nutzen öffentliche Mailboxen für ihren Informationsaustausch, der bei der Datenfernübertragung bekanntlich nicht an Landesgrenzen stößt. So können Computer-Anwender der verschiedensten Länder, ja sogar Kontinente, ihre Daten und Erfahrungen austauschen, und was das tollste an der ganzen Sache ist – man ist bei der Datenfernübertragung nicht auf ein bestimmtes Computer-System angewiesen! Diese Tatsache schafft völlig neue Kommunikationswege, bei entsprechender Gerätekonfiguration (Terminalprogramm, Akustikkoppler, Telefon etc.) sind alle Computer-Besitzer der Welt (!) untereinander verbunden und können miteinander kommunizieren. Diese neuen Kommunikationswege stehen zwar in ihrer Entwicklung noch am Anfang, doch scheint sich beim Anwender der Trend zur ernsthaften Nutzung dieser Systeme durchzusetzen.

Nach unserer Titelgeschichte von Heft 7/85 erreichte uns ein zweites Datenkommunikations-System für den Schneider CPC – Teleterminal 300s. Erstaunlich war zunächst der Preis von 178,– DM – der deutlich unter dem lag, was bisher an vergleichbaren Produkten vorlag.

In diesem Paket ist neben der Terminalsoftware auf Kassette und dem Bedienungsheft ein Interfacekabel enthalten – dieses System funktioniert also ohne Hardware-Erweiterung. Das Interfacekabel wird an den Druckerport des CPC angeschlossen und kann mit jedem handelsüblichen Akustikkoppler betrieben werden.

Die Software wird von Kassette sehr

schnell geladen und automatisch gestartet. Auf dem Bildschirm werden 24 Zeilen à 80 Zeichen dargestellt. Zusätzlich befindet sich am oberen Bildschirm die Statuszeile, an der jederzeit der Betriebszustand des Terminalprogramms abgelesen werden kann. Sehr übersichtlich ist die Bedienerführung. Per Tastendruck kann der Betriebszustand geändert werden und die durchgehende Menueführung bewahrt die Übersicht der doch reichhaltigen Funktionen. Durch diese Tatsache kann sich auch ein Neuling bzw. Anfänger sehr schnell in das Programm einarbeiten. Im Betrieb erwies sich das Teleterminal 300s als sehr zuverlässig. Selbst Systemänderungen (wie z.B. von Full und Half-Duplex etc.) gehen während der Kommunikation problemlos vonstatten.

Da das mitgelieferte Kabel am Druckerport angeschlossen wird, ist der gleichzeitige Betrieb von Terminal und Drucker nicht möglich. Dieses Manko wird jedoch durch einen internen Textspeicher wettgemacht, der bis zu 50 Bildschirmseiten speichern kann. Zusätzlich stehen noch acht Zwischenspeicher zur Verfügung, die Kurzinfos wie Telefonnummern, Adressen usw. aufnehmen können. Um die zwischengespeicherten Daten dennoch auszudrucken, kann während des Betriebes der Interfacestecker abgezogen und das Druckerkabel an den Computer angeschlossen werden – da am Druckerport des CPC in diesem Fall keine Spannung anliegt, ist dies ohne Systemabsturz und Datenverlust möglich.

Das beiliegende Bedienungsheft gibt über die wesentlichen Funktionen Auskunft. Allerdings stellte sich das auf der Verpackung angekündigte 32-seitige, deutsche Handbuch als 15-seitiges Heftchen heraus. Dieser Umstand kann allerdings die gute Qualität der Kommunikations-Software nicht beeinträchtigen. Ein weiterer Pluspunkt ist die Kompatibilität und automatische Erkennung des Kassetten- oder Diskettenbetriebes – das Programm läuft also ohne Änderung mit Kassettenlaufwerk und Disk.

0711/432197
Schneider-User-Gruppe im BNTe.V.
06131/43309
Schneider-Mailbox FCBGM von 19.00 - 23.00 Uhr, Stop-bit 1, keine Parität
06081/9677
Taunus-Mailbox
069/63501110-29
Control Data
030/3052635
Berliner-Mailbox, 18.00 - 9.00 Uhr
069/494201
A.U.G.E (Apple-User-Groupe Europe), 19.00 - 6.00 Uhr
030/8922239
Daily Mail, 20.00 - 7.00 Uhr
07031/278296
Elias, 17.30 - 22.00 Uhr

Fazit:

Für den Preis von ca. DM 178,– (ohne Koppler) erhält man ein Kommunikationssystem, das den üblichen Anforderungen entspricht. Die am Druckerport simulierte RS-232-Schnittstelle erfüllt ihren Zweck, der gleichzeitige Druckerbetrieb wird bei 50 Seiten Textspeicher wohl nur selten vermißt. Die Software besticht durch gutes Handling, Übersichtlichkeit und nicht zuletzt durch eine schnelle Abarbeitungsgeschwindigkeit.

Insgesamt stellt Teleterminal 300s eine interessante Alternative zu Kommunikationssystemen mit Hardware-Erweiterung dar.

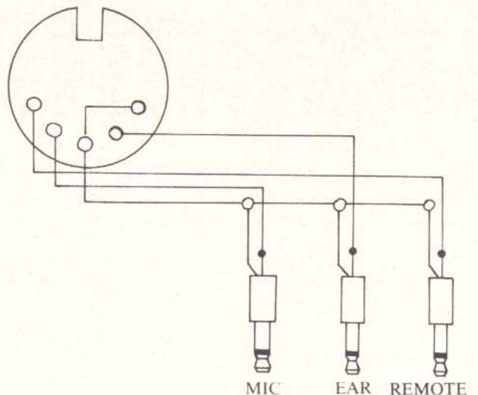
Hersteller und Vertreiber von Teleterminal 300s ist die Fa. Microcomputer in Berlin. Das System ist ebenfalls bei der Fa. Janke, Wuppertal erhältlich. (SR)

Kassettenanschluß für CPC 664 / 6128



Leider ist das Angebot von Software auf Disketten noch immer sehr gering. 90 % der Spielesoftware ist nur auf Kassette erhältlich. Was macht nun der 664 Anwender, der keinen Datacorder in seinem Rechner eingebaut hat. Laut Handbuch braucht er lediglich einen handelsüblichen Kassettenrecorder über ein DIN-Kabel mit dem Rechner zu verbinden und schon kann er auch diese Software nutzen. Leider haben jedoch die wenigsten Kassettenrecorder einen genügend hohen Ausgangspegel an der DIN-Buchse anliegen. Der Rechner kann die Daten aufgrund der niedrigen Lautstärke nicht verstehen. Abhilfe schafft ein kleiner Vorverstärker, der in das Kabel eingeschaltet wird. Dies ist zwar die eleganteste, jedoch auch die komplizierteste und teuerste Lösung.

Wesentlich einfacher und billiger geht es, wenn man den CPC anstelle mit der DIN-Buchse an den Kopfhörer und Mikrofonanschluß anschließt. Hier sind nämlich schon die Vorverstärker eingebaut. Datenrecorder, die heute schon für unter 100,– DM erhältlich sind, haben sogar einen Filter für Rechtecksignale eingebaut. Hat der Recorder einen Remote-Eingang, so läßt sich sogar noch die Motorsteuerung vom Rechner übernehmen. Bei nahezu allen Billigrecordern, die für diese Aufgabe am besten geeignet sind, bestehen diese Eingänge aus zwei 1,55 mm Klinken-



buchsen. Die Remote-Kontrolle ist ein 0,7 mm Klinkenstecker.

Verbinden Sie die Bauteile:

2 Klinkenstecker 1,5 mm
1 Klinkenstecker 0,7 mm
1 DIN-Stecker 5-polig
ca. 1 m abgeschirmtes Kabel 6 adrig

laut Zeichnung miteinander, adjustieren Sie die Lautstärke Ihres Recorders und schon ist das Problem des Kassetteninterfaces gelöst.

(TM)



Hier sind sie nun, die Gewinner unserer Fragebogenaktion aus Heft 6/85! Überrascht waren wir wegen der gewaltigen Resonanz, denn allein in den ersten beiden Wochen nach Erscheinen unserer Zeitschrift gingen mehrere Tausend beantwortete Fragebögen in unserer Redaktion ein.

An dieser Stelle möchten wir allen unseren treuen Lesern ein herzliches Dankeschön sagen. Die aus der Befragung gewonnenen Erkenntnisse wurden bereits in unsere „CPC International" eingearbeitet.

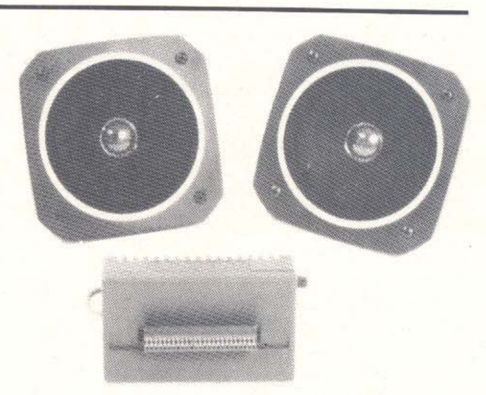
Die Gewinne werden in den nächsten Tagen zugestellt.

1. Preis

Thorsten Elge, Berlin

2. – 10. Preis

Girici Belgin, Freigericht
Rico Nobis, Mielkendorf
Konrad Wanner, Wedemark
Yildirim Itzeln, Krefeld
Peter Pesch, Haan
Horst Schrödter, Bielefeld
Horst Hauschildt, Talkau
Frank-Udo Hoffmann, Jülich
Gaby Lausberg, Ludwigshafen



11. – 20. Preis

Jürgen Kreuzberger, Sauerthal
Andreas Gugger-Bruni, Uetendorf
Reinhard Liesegang, Wilhelmshaven
Michael Molnar, Villingen
Gisbert Rostock, Berlin
Georg Hey, Schönenbeck
Wolfgang Borsetzky, Lemgo
Peter Krick, Sulzbach
Norbert Gabel, Eichstätt
Hans-Jürgen Hubert, Bad Krozingen

Neue Wege in der Werbung

Vor dem Kauf von Computerprodukten, dies trifft vor allem für Software zu, möchte sich der User naturgemäß von der Beschaffenheit und Qualität der angebotenen Produkte informieren.

In Zusammenarbeit mit der V.I.T. Bertelsmann Video GmbH beschreitet Data Media Software jetzt einen neuen Weg in der Präsentation von Software. In den Computerabteilungen der Kaufhäuser Hertie, Karstadt und Brinkmann werden in Kürze Video-Clips vorgeführt, die Informationen über Soft- und Hardware aus dem Data Media Angebot beinhalten. Nach einem Zeitraum von 6 Wochen werden die Videos ausgetauscht und aktualisiert.

Data Media beabsichtigt, auch weitere Händler an dieses System anzuschließen. Ansprechpartner ist Herr Freier, Data Media, Ruhrallee 55, 4600 Dortmund, Telefon: 0231/12 50 71.

Z-80 Assemblerkurs

Nachdem nun im letzten Teil unseres Kurses die Addition zweier 16-Bit-Zahlen behandelt wurde, wollen wir uns in diesem Teil mit der binären Multiplikation befassen.

Multiplikation durch fortgesetzte Addition:

Die langsamste Art, zwei ganze Zahlen zu multiplizieren besteht darin, den Multiplikanden so oft zu 0 zu addieren, wie der Multiplikator angibt. Ein kleines Beispiel:
geg.:
Operand 1: 121
Operand 2: 7

Lösung 1:
121 + 121 + 121 + 121 + 121 + 121 + 121 = 847
Lösung 2:
7 + 7 + 7 + 7 + 7 + (insgesamt 121 mal) = 847

Die fortgesetzte Addition kann optimiert werden, indem zuerst festgestellt wird, welcher Operand sich besser zum Multiplikator und welcher sich besser zum Multiplikand eignet. Dazu genügt ein einfacher Größenvergleich. Außerdem sollte bei der Programmierung in Assembler die Größe des höchst möglichen Produktes beachtet werden. Ist ein Operand 8 Bits, dann kann das Ergebnis höchstens (255 * 255 =) 65025, also eine 16-Bit-Zahl ergeben.

Hier nun das entsprechende Maschinenprogramm:

```
 10         ORG  32768                   ;Startadresse 32768
 20         LD   DE,(OP1)    ED 5B 12 80 ;DE <- OP1 UND OP2
 30         LD   B,D         42          ;OP2 ins Zaehlregister
 40         XOR  A           AF          ;Loeschen von A
 50         LD   D,A         57          ;geloescht wird D-Register (MSB von DE)
 60         LD   L,A         6F          ;geloescht wird L-Register
 70         LD   H,A         67          ;geloescht wird H-Register
 80         CP   B           67          ;ist OP2 gleich Null?
 90         JR   Z,FERTIG    6F          ;wenn ja, dann Sprung nach FERTIG
100 SCHLEI  ADD  HL,DE       19          ;OP1 zu HL addieren
110         DJNZ SCHLEI      10 FD       ;solange zurueck, bis B-Register = 0
120 FERTIG  LD   (PRO),HL    22 14 80    ;Ergebnis in PRO abspeichern
130         RET              C9          ;zurueck zum aufrufenden Programm
140 OP1     DEFB 1           00          ;1-Byte-Variable   Multiplikator
150 OP2     DEFB 1           00          ;1-Byte-Variable   Multiplikant
160 PRO     DEFB 2           00 00       ;2-Byte-Variable   Produkt
170         END
```

Sie haben vielleicht bemerkt, daß die Programme allmählich länger werden, und damit verbunden geht der Überblick verloren. Deshalb ist es sehr wichtig, daß wir, bevor wir uns an den Computer setzen und etwas ausprobieren (Gefahr des Absturzes!), den Ablauf festlegen. Dazu sollte es jedem selbst überlassen werden, auf welche Weise er sich Notizen macht. Es ist auf jeden Fall gut, die allgemein anerkannten Symbole zu verwenden; es bieten sich das Struktogramm (Nassi Schneidermann) oder ein Programmablaufplan (auch Flußplan genannt) an.

Damit Schneider Besitzer, die keinen Assembler besitzen, auch dieses Programm fahren können, das äquivalente Basic-Programm:

```
100 REM Multiplikation durch fortgesetzt
e Addition
110 REM
120 ANF = 32768
130 ENDE = ANF + 17
140 OP1 = ENDE + 1
150 OP2 = ENDE + 2
160 PRO = ENDE + 3
170 REM
180 REM Maschinenprogramm abspeichern
190 REM
200 MEMORY 32767
210 FOR I=ANF TO ENDE
220 READ A$
230 POKE I,VAL("&"+A$)
240 NEXT I
250 DATA ED,5B,12,80,42,AF,57,6F,67,67,6
F,19,10,FD,22,14,80,C9
260 REM
270 REM Eingabe der 8-Bit-Zahlen
280 REM
290 INPUT "OPERAND 1";A
300 IF A>255 OR A<0 THEN PRINT "Zahl aus
serhalb des gueltigen Wertebereiches!":G
OTO 290
310 POKE OP1,A
320 INPUT "OPERAND 2";A
330 IF A>255 OR A<0 THEN PRINT "Zahl aus
serhalb des gueltigen Wertebereiches!":G
OTO 290
340 POKE OP2,A
350 CALL &8000
360 PRINT "Das Produkt ist ";PEEK (PRO)
+ 256 * PEEK (PRO+1)
370 GOTO 290
```

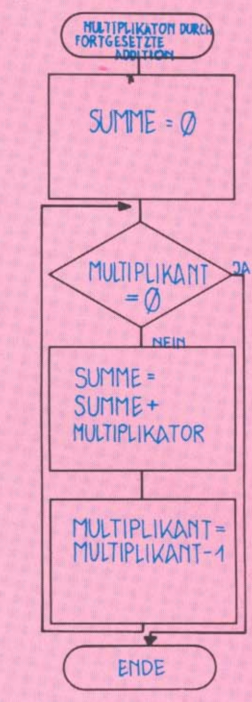
Wenn wir nun dieses Verfahren auf größere Zahlen anwenden, dann bekommt man die Grenzen der Leistungsfähigkeit eines Z-80-Prozessors zu spüren, und dies, obwohl der Prozessor des CPC etwa 825000 einfache Maschinencode-Instruktionen in der Sekunde durchführen kann. Würden wir 3-Byte-Zahlen miteinander multiplizieren, so könnte mit Wartezeiten von mehreren Sekunden gerechnet werden. Man ließ sich also etwas anderes einfallen. Zum Erklären eignet sich die Methode, die wir in der Schule gelernt haben – schriftliches Multiplizieren:

Schriftliches Multiplizieren:

Um den Algorithmus für eine duale Multiplikation zu entwickeln, wollen wir damit beginnen, eine gewöhnliche dezimale Multiplikation zu untersuchen. Wir wollen 14 und 27 schriftlich multiplizieren:

```
 14 * 27 (Multiplikator und Multiplikant)
---------
 28      (Teilprodukt)
  98     (Teilprudukt)
---------
 378     (Endprodukt)
```

Flußplan 1



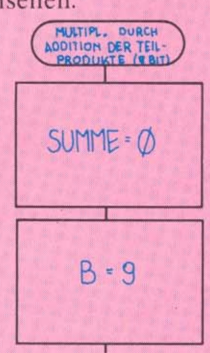
Die Multiplikation wird ausgeführt, indem man die linke Ziffer des Multiplikators mit dem Multiplikanden multipliziert, d.h., 2 * 14. Das Teilprodukt ergibt 28. Danach wird das Ergebnisfeld um eine Stelle nach links geshiftet und man addiert das zweite Teilprodukt, 7 * 14 = 98. Als Ergebnis erhalten wir dann 378. Die duale Multiplikation wird auf die gleiche Weise ausgeführt.

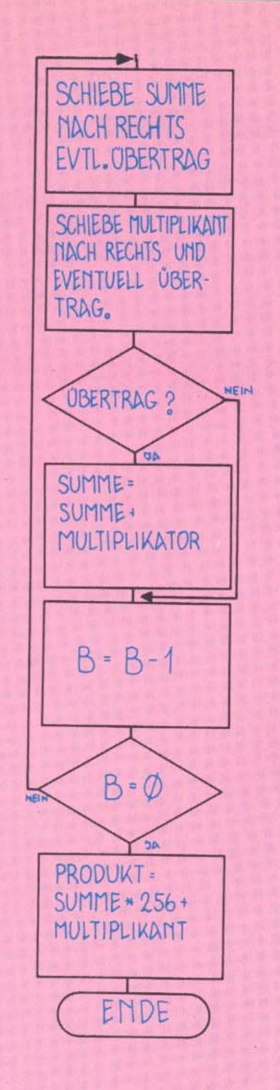
Duale Multiplikation:

An einem Beispiel ist dies leicht zu erklären. Es sollen die Zahlen 9 und 13 miteinander multipliziert werden:

```
1001 * 1101 (Multiplikator und Multiplikant)
------------
1001        (Teilprodukt 1)
 1001       (Teilprodukt 2)
  0000      (Teilprodukt 3)
   1001     (Teilprodukt 4)
------------
1110101     (Endprodukt = 117)
```

Bevor wir nun loslegen, um diese Art der Multiplikation zu programmieren, wollen wir uns den Algorithmus anhand eines Struktroprogrammes genauer ansehen.





Da das Produkt von zwei 8-Bit-Zahlen 16 Bit breit werden kann, muß ein Doppelregister als Zähler dienen:

```
 10             ORG  32768                        ;Startadresse 32768
 20             LD   HL,OP1       21 17 80        ;Die Adresse des Operanden nach HL
 30             LD   D,(HL)       56              ;D-Register wird mit OP1 geladen
 40             INC  HL           23              ;HL-Register +1 = Adresse von OP2
 50             LD   C,(HL)       4E              ;C-Register wird mit OP2 geladen
 60             LD   B,9          06 09           ;Schleife initialisieren
 70             XOR  A            AF              ;Akku loeschen
 80 SCHLEI      RRA               1F              ;A - Rechts rotieren durch Carry
 90             RR   C            CB 19           ;Carry in C-Register rollen
100             JR   NC,WEITER    30 01           ;Wenn Carry, dann addieren
110             ADD  A,D          82              ;Addition der Teilprodukte
120 WEITER      DJNZ SCHLEI       10 F8           ;8 Bits + Sonderfall
130             LD   B,A          47              ;Highbyte des Produktes in B
140             LD   (PRO),BC     ED 43 19 80     ;Ergebnis abspeichern
150             RET               C9              ;Zurueck zum aufrufenden Programm
160 OP1         DEFB 1            00              ;Operand 1
170 OP2         DEFB 1            00              ;Operand 2
180 PRO         DEFB 2            00              ;Produkt
190             END
```

```
100 REM Multiplikation durch Addition de
r Teilprodukte
110 REM
120 ANF = 32768
130 ENDE = ANF + 22
140 OP1 = ENDE + 1
150 OP2 = ENDE + 2
160 PRO = ENDE + 3
170 REM
180 REM Maschinenprogramm abspeichern
190 REM
200 MEMORY 32767
210 FOR I=ANF TO ENDE
220 READ A$
230 POKE I,VAL("&"+A$)
240 NEXT I
250 DATA 21,17,80,56,23,4E,06,09,AF,1F,C
B,19,30,01,82,10,F8,47,ED,43,19,80,C9
260 REM
270 REM Eingabe der 8-Bit-Zahlen
280 REM
290 INPUT "OPERAND 1";A
300 IF A>255 OR A<0 THEN PRINT "Zahl aus
serhalb des gueltigen Wertebereiches!":G
OTO 290
310 POKE OP1,A
320 INPUT "OPERAND 2";A
330 IF A>255 OR A<0 THEN PRINT "Zahl aus
serhalb des gueltigen Wertebereiches!":G
OTO 290
340 POKE OP2,A
350 CALL &8000
360 PRINT "Das Produkt ist ";PEEK (PRO)
+ 256 * PEEK (PRO+1)
370 GOTO 290
```

Beschreibung:

Zeile 10 - ORG 32768
Gibt dem Assembler an, ab welcher Stelle das erzeugte Maschinenlisting einmal abgespeichert wird.
Zeile 20 - LD HL, OP1
Die Adresse des ersten Operanden wird in HL gespeichert. Dadurch wird ein schnellerer Ablauf erreicht.
Zeile 40 - INC HL
Das Doppelregister L wird inkrementiert, d.h., es wird um eins erhöht.

Dadurch zeigt HL auf OP2.
Zeile 30 - LD D, (HL)
Das Register D wird mit dem ersten Operanden geladen.
Zeile 50 - LD C, (HL)
Das Register C wird mit dem zweiten Operanden geladen.
Zeile 60 - LD B,9
Da das Produkt vor der Addition der Teilprodukte rotiert wird, muß am Ende der Additionen das Produkt noch einmal rotiert werden. Dies läßt sich sehr einfach umgehen, indem wir das B-Register, das ja zur Zählung benutzt wird, mit 9 anstatt mit 8 initialisieren.
Zeile 70 - XOR A
Eine einfache binäre Exklusiv-Oder-Verbindung zwischen Akku und Akku.

Dabei kommt natürlich immer Null heraus. Dies ist also ein Löschbefehl für den Akku.

Zeile 80 - SCHLEI RRA
Das SCHLEI ist ein Label und dient dem Programmierer zur einfacheren Eingabe. Der Assembler setzt nach einer Definition automatisch bei Benutzung eines Labels dessen Adresse ein.

Das RRA ist ein neuer Befehl. Er wurde eigentlich aus Kompatibilitätsgründen zum 8080-Prozessor im Z-80 verwendet. Den gleichen Befehl kann man auch RRA schreiben. Dann allerdings wird Ihr Assembler einen anderen Byte-Code verwenden. Im Gegensatz zum 8080 ist es möglich, alle Allzweckregister zu rotieren oder zu shiften.

Beim 8080 konnte dies nur der Akku. Bedeutung: Der Inhalt des Akkumulators wird rechts durch das Carry-Flag gerollt, d.h.:

```
Bit 7 --> Bit 6
Bit 6 --> Bit 5
Bit 5 --> Bit 4
Bit 4 --> Bit 3
Bit 3 --> Bit 2
Bit 2 --> Bit 1
Bit 1 --> Carry
Carry --> Bit 7
```

Zeile 90 - RR C
Die Bedeutung dieses Befehls ist fast die gleiche, mit dem Unterschied, daß diesmal statt des Akkus das C-Register angesprochen wird.
Zeile 100 - JR NC,WEITER
Hier fällt die Entscheidung, ob der Operand 2 addiert wird, oder ob die Addition übersprungen wird.
Zeile 110 - ADD A,D
Diese Stelle wird nur erreicht, wenn das Carry im letzten Befehl gesetzt war. Die eigentliche Addition der Teilprodukte wird vorgenommen.
Zeile 120 - DJNZ SCHLEI
Der Schleifenzähler, das B-Register, wird dekrementiert (B=B-1). Danach wird geprüft, ob B den Wert Null hat. Sollte dies nicht der Fall sein, so findet ein Sprung zu der angegebenen Adresse (=SCHLEI) statt. Im anderen Fall wird einfach die sequenzielle Abarbeitung fortgesetzt.
Zeile 130 - LD B,A
Eine einfache Register-Register-Operation: Der Akku enthält nach Austritt aus der Schleife, das Highbyte des Produktes. Nach dem Umladen hat man das Ergebnis im BC-Register als eine 16-Bit-Zahl vorliegen.
Zeile 140 - LD (PRO), BC
Das Doppelregister BC wird im Speicher abgelegt. Dabei ist nach Ausführung des Programms in der Speicherstelle PRO das C-Register, in der darauf folgenden PRO+1 das B-Register.
Zeile 160 - RET
Diese Zeile veranlaßt die Rückkehr zum aufrufenden Programm. Dies kann sowohl ein Maschinen- als auch ein Basic-Programm sein.
Zeile 170 - OP1 DEFB 1
In dieser Zeile wird eine Variable definiert, die ein Byte lang ist und dessen Adresse OP1 genannt wird.
Zeile 180 - OP2 DEFB 1
Es gilt das gleiche wie in Zeile 170.
Zeile 190 - PRO DEFB 2
Hier wird eine Zwei-Byte-Variable definiert. Ihre Adresse kann mit PRO erreicht werden.
Zeile 200 - END
Dies ist, wie die meisten wissen, eine Assemblerdirektive. Durch dieses Codewort weiß der Assembler, daß das Programm hier zu Ende ist.
Ich hoffe, Sie haben alles verstanden. Leider ist es manchmal schwierig, gewisse Algorithmen zu verstehen. Da dies aber Grundlagen der Programmierung in Assembler sind, sollte man sich bemühen, die Vorgehensweise zu verstehen. Im nächsten Teil lernen Sie etwas über die Firmware, d.h., die Betriebssystemroutinen kennen, wie man sie anwendet und was dabei zu beachten ist. (HF)

Rechengenauigkeit



Herr Ernst Hügli aus Kestenholz/Schweiz schreibt uns:
Ich beziehe mich auf den Leserbrief von Herrn Jürgen Otto, Hamburg und Ihre Antwort darauf, die in der Leserbriefecke von CPC International Nr. 6 (Juni 1985) erschienen ist. Diese Antwort ist, gelinde gesagt, etwas befremdlich. Natürlich ist mit Ihrer Lösung das lästige Problem vom Bildschirm verschwunden. Dafür rumort es jetzt unerkannt im RAM und dürfte jede Menge Ärger verursachen, wenn es Herrn Otto einfallen sollte, mit der Summe von n Summanden 0.01 noch etwas anzufangen.
Ihr Leser ist hier auf ein bekanntes Problem gestoßen: INTEGER-Zahlen können korrekt dargestellt werden, sofern sie aus dem Bereich ±32'767 stammen. INTEGER-Zahlen, die dem Betrage nach größer sind, oder REAL-Zahlen können in der Regel nicht mehr korrekt (d.h. mit allen Ziffern) wiedergegeben werden. Der Schneider CPC 464 „spendiert" für solche Zahlen einen Speicherbereich von 5 Byte: vier Byte für die sog. Mantisse (inkl. Vorzeichen) und eines für den Exponenten (ebenfalls inkl. Vorzeichen). Mit dem folgenden Programm können wir diese fünf Bytes herauslesen:

```
DAS ZEICHEN § ENTSPRICHT DER TASTE 26 IH
RES CPC (KLAMMERAFFE)

100 REM ******************************
110 REM * RECHENGENAUIGKEIT CPC 464 *
120 REM ******************************
130 REM
140 REM COPYRIGHT BY E. HUEGLI
150 REM 1985-06-09; 2. VERSION
160 REM
170 CLS:A=0:MAXIMUM=100: 'ANZAHL WIEDERH
OLUNGEN
180 FOR I = 1 TO MAXIMUM
190     A=A+0.01: 'RECHENMETHODE NACH J.
OTTO
200     B=1/100 : 'BESSERE RECHENMETHODE
210     PRINT A,B:'AUSDRUCK BEIDER VARIAN
TEN ZUM VERGLEICH
220     ADR1=§A:ADR2=§B:'VAIABLENADRESSEN
 IM RAM
230     PRINT PEEK(ADR1);PEEK(ADR1+1);PEE
K(ADR1+2);PEEK(ADR1+3);PEEK(ADR1+4): 'SP
EICHERINHALT AUSDRUCKEN
240     PRINT PEEK(ADR2);PEEK(ADR2+1);PEE
K(ADR2+2);PEEK(ADR2+3);PEEK(ADR2+4): 'SP
EICHERINHALT AUSDRUCKEN
250     PRINT
260 NEXT I
270 END
```

Der Trick mit dem „Klammeraffen" @, der uns die Adresse einer Variablen im RAM vermittelt, kann im Buch „CPC 464 Intern" nachgelesen werden.
Lassen wir das Programm laufen, dann erhalten wir folgenden Ausdruck auf dem Monitor-Schirm (Auswahl):

```
Ø.Ø1                 Ø.Ø1                        Ø.Ø2              Ø.Ø2
61  1Ø  215  35  122                             61  1Ø  215  35  123
61  1Ø  215  35  122                             61  1Ø  215  35  123

Ø.Ø3                 Ø.Ø3                        Ø.Ø4              Ø.Ø4
92  143  194  117  123                           61  1Ø  215  35  124
92  143  194  117  123                           61  1Ø  215  35  124

Ø.Ø5                 Ø.Ø5                        Ø.65              Ø.65
2Ø4  2Ø4  2Ø4  76  124          ....             1Ø4  1Ø2  1Ø2  38  128
2Ø5  2Ø4  2Ø4  76  124                           1Ø2  1Ø2  1Ø2  38  128

Ø.66ØØØØØØ1          Ø.66                        1                 1
145  194  245  4Ø  128          ....             1  Ø  Ø  Ø  129
143  194  245  4Ø  128                           Ø  Ø  Ø  Ø  129
```

Durch Abändern der Variablen „maximum" in Zeile 170 können wir für weitere Werte den Ausdruck bekommen:

```
1.Ø2                 1.Ø2                        1.Ø4              1.Ø4
41  92  143  2  129             ....             81  184  3Ø  5  129
41  92  143  2  129                              82  184  3Ø  5  129

1.2                  1.2
145  153  153  25  129
154  153  153  25  129
```

Doch jetzt zur Deutung dieses Ausdrucks:

Die ersten vier Zahlen sind die dezimal geschriebenen vier Bytes der Mantisse, wobei das Byte mit der niedrigsten Wertigkeit an erster Stelle steht. Die fünfte Zahl stellt den dezimal geschriebenen Zweierexponenten dar. Das 4. Byte (d.h. das höchstwertige Byte) enthält in verschlüsselter Form das Vorzeichen: Ist sein Wert (dezimal) kleiner als 128, dann ist die Mantisse positiv, sonst negativ. Damit opfert man in der binären Schreibweise zwar das 32. Bit, da man sich aber an die Konvention hält, daß REAL-Zahlen immer in normierter Form geschrieben werden, d.h. 1.XXXXXXXX......X x 2^{exp} (wo x für irgendwelche Binärziffern steht), ist das 32. Bit der Mantisse immer 1 und man braucht es nicht explizite zu schreiben. Das 5. Byte enthält nun auch nicht den „richtigen" Zweierexponenten exp, sondern den mit einem Offset von +129 versehenen Exponenten. 0 entspricht damit dem Zweierexponenten −129, 255 dem Exponenten +126. Mit diesen Erklärungen können wir die interne, binäre Darstellung der Zahlen rekonstruieren. Betrachten wir als Beispiel die Zahl 0.01; binär geschrieben lauten die vier Bytes der Mantisse:

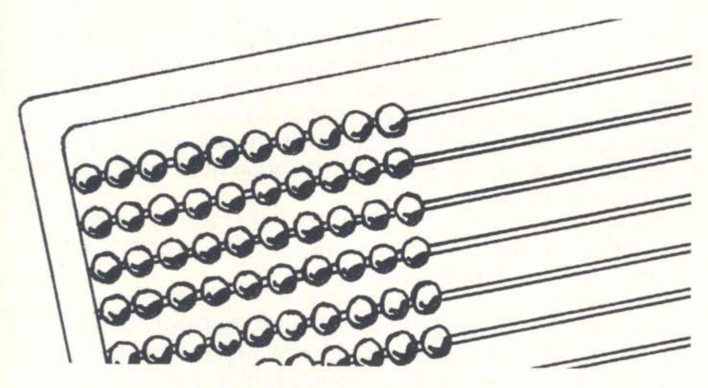
1. (niederwertigstes) Byte : 61 = 00111101
2. Byte : 10 = 00001010
3. Byte : 215 = 11010111
4. (höchstwertiges) Byte : 35 = 00100011
5. Byte für den Exponenten: 122 → exp = 122 − 129 = −7

Das vierte Byte ist kleiner als 128 (= 10000000), folglich ist die Mantisse positiv; die führende Null, die nur das Vorzeichen anzeigt, wird durch eine Eins ersetzt. Endgültig lautet damit diese Zahl:

0.01 = 1.0100011'11010111'00001010'00111101 x <2^{-7}>
bzw. in der normalen Dualbruchschreibweise:
0.01 = 0.00000010100011110101110000101000111101

was tatsächlich die binäre Darstellung von 0.01 ist. Wer bis hierher vorgestoßen ist, erkennt spätestens jetzt den Grund für die Probleme von Herrn Otto:

0.01 ist zwar in dezimaler Schreibweise ein endlicher, abbrechender Bruch, und wir können problemlos mit allen Stellen rechnen. In der dualen Schreibweise des Computers ist es aber ein rein periodischer, unendlicher Bruch (ähnlich wie z.B. 1/7 = 0.142857142857... ein unendlicher, periodischer Dezimalbruch ist), von dem wir grundsätzlich nicht alle Stellen berücksichtigen können. Das Abbrechen nach endlicher Stellenzahl führt aber zu Rundungsfehlern, was sich ab einem gewissen Moment auswirkt. Aus diesem Grund verfügen Taschenrechner über sog. Schutzstellen, d.h., sie rechnen intern mit mehr Stellen, als sie nachher anzeigen. Damit kann zwar das Problem der Rundung von Brüchen nicht umgangen werden, aber die Anzeige täuscht dann genaueres Rechnen vor (so wie der „Trick" mit dem PRINT USING, mit dem man das Problem zwar verdrängen, aber eben nicht lösen kann).



Um nun sichtbar zu machen, von welchem Augenblick an sich die Rundungsfehler, die beim Aufaddieren von 0.01 entstehen, bemerkbar machen, wurden die Zeilen 200, 210 und 240 ins Listing eingefügt. Zeile 200 zeigt eine bessere Lösung des Problems von Herrn Otto: Durch die For-Next-Schleife wird eine INTEGER-Zahl i aufsummiert (im zur Diskussion stehenden Zahlenbereich ein unproblematisches Vorhaben) und anschließend durch 100 dividiert. Zeile 210 zeigt den fortlaufenden Wert (auch ohne PRINT USING immer das, was wir erwarten) und Zeile 240 den entsprechenden RAM-Inhalt. Und da zeigt es sich nun, daß bereits nach 5 Additionen ein Unterschied zwischen 0.01+0.01+0.01+0.01+0.01 und (1+1+1+1+1)/100 besteht:

0.000011001100110011001100110011001100
bzw.
0.000011001100110011001100110011001101

Der Unterschied von einer Einheit in der 36. Stelle nach dem Komma ist so gering, daß ihn der CPC 464 (noch) nicht darstellen kann: Die Zeile 210 liefert für beide Variablen den Ausdruck 0.05. Wird aber anschließend ein Test auf Gleichheit der bei Zahlen durchgeführt, etwa

```
215 IF a = b THEN PRINT "a und b sind gleich" ELSE
PRINT "a und b sind verschieden"
```

so zeigt sich, daß der BASIC-Interpreter aufgrund dieser einen Stelle, in der die beiden Zahlen sich unterscheiden, die beiden Zahlen nicht mehr als gleichwertig anerkennt.

Damit wären wir bei der Rechtfertigung angelangt, warum wir für dieses Problem so viel Papier verbraucht haben: Naiverweise werden immer wieder REAL-Zahlen auf Gleichwertigkeit geprüft – nicht nur von Anfängern, denn oftmals realisiert man gar nicht, was man eigentlich macht, wie etwa im folgenden kleinen Programm:

```
100 a = 0: 'initialisieren
110 WHILE a  < >1
120 a = a + 0.01
130 PRINT a : PRINT
140 WEND
```

Mit dieser Programmier-Methodik hat man sich – vermutlich ohne es zu wollen – in einer unendlichen Schleife gefangen, denn die Abbruchbedingung a = 1 wird nie erfüllt werden; es bleibt nur noch die Notbremse BREAK. Solche Situationen führen dann zur Forderung, Tests auf Gleichheit dürften nur mit INTEGER-Zahlen im korrekt darstellbaren Bereich durchgeführt werden. Aus dem gleichen Grund ist es in der Programmiersprache Pascal unmöglich, in einer FOR-Schleife einen REAL-Wert als Schrittweite zu benutzen, was das "unsaubere" BASIC bekanntlich zuläßt:

```
110 FOR a = 0 TO 1 STEP 0.01
    .
    .
    .
140 NEXT a
```

ist sehr wohl möglich im voranstehenden Programm – und prompt stoppt seine Ausführung bei 0.990000001, und nicht wie verlangt bei 1; Gründe dafür siehe oben.

Fazit dieser Darstellungen:

1) Außer bei INTEGER-Zahlen im Bereich ±32'767 verzichtet man aus Prinzip auf den Test auf Gleichheit.

2) Man mißtraue aus Prinzip den auf dem Bildschirm (oder auf dem Drucker) ausgegebenen Zeilen.

Ernst Hügli

Steuerzeichen des CPC 464 für Bildschirm und Drucker Teil I

1. Bildschirmausgabe

Man könnte als Überschrift auch schreiben: „Geheimnisse des CPC", denn für „Einsteiger" bleiben die Steuerzeichen des CPC ein Geheimnis. Zwar werden im Bedienungshandbuch zum CPC im Kapitel 1, Seite 7, diese (nicht druckbare) Zeichen erwähnt, jedoch mit dem Hinweis, man solle sich vorerst nicht damit beschäftigen.
Später wird in dem Handbuch im Kapitel 9.1 etwas eingehender darüber geschrieben. Dort werden die Steuerzeichen auch „Kontrollzeichen" genannt. Leider sind diese Erläuterungen für einen Einsteiger ungenügend, weil erklärende Beispiele fehlen. Dem soll mit diesem Artikel abgeholfen werden.

In Listings, die in Zeitschriften und Büchern abgedruckt sind, wird man diese Steuerzeichen auch nie finden. Der Grund ist der, daß diese Zeichen, vom Listing ausgehend, sich nicht ausdrucken lassen. Normalerweise kann man sie nur in Programmzeilen eingeben. Sie verschwinden aber vom Bildschirm oder erscheinen nicht im gedruckten Text, sobald die Programmzeile vom Computer ausgeführt worden ist. Es gibt aber eine Möglichkeit, diese Steuerzeichen mit folgendem Programm auf dem Bildschirm auszugeben. Vorher aber noch eine Bemerkung. Wenn in den folgenden Programmzeilen ein unterstrichenes Zeichen aufgeführt ist, so bedeutet es, daß das entsprechende Steuerzeichen mit der Taste CTRL und der Taste, die das unterstrichene Zeichen hat, einzugeben ist. Nun unser Programm:

```
10 MODE 1:CLS:i=0
20 FOR n=1 TO 7
30 PRINT
40 FOR i=i TO i+4:PRINT i"=A"CHR$(i)"   ";
50 IF i=31 THEN 90
60 NEXT i
70 PRINT
80 NEXT n
90 PRINT:PRINT
```

Die Darstellung der Zeichen durch den Drucker ist nur möglich, wenn man eine „Screen Dump"-Kopie vom Bildschirm macht, wie nachstehend abgebildet. Dort sind aufgeführt die Werte der Zeichen, die Abbildung der Steuerzeichen und die Buchstaben (oder Zeichen) der Tasten, mit denen sie, mittels CTRL Taste, aufgerufen werden können.

Wert	0	1	2	3	4	5	6	7	8	9	10
Zeich.	□	┌	⊥	┘	ϟ	⊠	✓	Ω	←	→	↓
Taste	( )	A	B	C	D	E	F	G	H	I	J

Wert	11	12	13	14	15	16	17	18	19	20	21	22	23
Zeich.	↑	↓	↵	⊗	⊙	⊟	◷	◶	◵	◴	✕	Π	⊣
Taste	K	L	(M)	N	O	(P)	Q	R	S	T	U	V	W

Wert	24	25	26	27	28	29	30	31
Zeich.	⧗	♦	?	⊖	◰	◱	◲	◳
Taste	X	Y	Z	[	\	]	↑	@

Wir können jetzt im Handbuch des CPC im Kapitel 9, Seiten 2 bis 4, die Angaben ergänzen, indem wir vor die einzelnen Werte jeweils den Buchstaben (oder das Zeichen) der entsprechenden Taste und die Abbildung des Steuerzeichens eintragen. Damit wird uns die Tabelle schon verständlicher. Jedes entsprechende Steuerzeichen im Text oder in einer PRINT-Anweisung löst die rechts in der Tabelle beschriebene Funktion aus.
Wie wir feststellen werden, kann man die Zeichen mit dem Wert 0, 13 und 16 nicht mit den Tasten eingeben. Bei 13 (Taste M) wird der Cursor gleich eine Zeile nach unten gesetzt. Bei 16 (Taste P) hört man einen Ton, sonst scheint aber nichts zu geschehen. Ob und wie das Zeichen mit dem Wert 0 eingegeben werden kann, ist dem Autor nicht bekannt. Das Zeichen mit dem Wert 9 ist identisch mit dem der Taste TAB (unter der Taste ESC).

Grundsätzlich ist es so, daß alle Steueranweisungen in Listings, die in der Form **PRINT CHR$(n)** enthalten sind und bei denen der Wert n zwischen 1 und 31 liegt, durch entsprechende Steuerzeichen mit dem gleichen Wert n ersetzt werden können, bis auf die mit den Werten 13 und 16. Der Befehl heißt dann **PRINT "n"** (n=Steuerzeichen). Das Steuerzeichen wird in Anführungsstriche gesetzt und kann auch mitten in einem Text eingebaut sein.

Wir wollen uns jetzt aus der Liste des Handbuchs, Kapitel 9.1, einige Anwendungen herausgreifen und Beispiele geben. Wie wir das Steuerzeichen mit dem Wert 1 benutzen können, haben wir weiter oben bereits gesehen. Weil wir dort eine Schleife hatten, wurde das auszugebende Zeichen mit CHR$(i) programmiert. Wenn wir ein einzelnes Zeichen auf dem Bildschirm ausgeben wollen, können wir auch eingeben:

PRINT "AJ"

J steht für den Pfeil nach unten, der dann auch auf dem Bildschirm erscheint. Das Steuerzeichen davor hat nur die Funktion, das Zeichen auf dem Bildschirm auszugeben, die Funktion dieses Steuerzeichens aber zu unterbinden.
Im nächsten Beispiel soll auf einen Fehler hingewiesen werden durch Text und Warnton. Wir zeigen zwei Zeilen, eine mit CHR$ und eine mit Steuerzeichen:

```
10 PRINT CHR$(7)"Achtung Fehler!"
20 PRINT "GAchtung Fehler!"
```

In der Zeile 20 brauchte man nur eine Taste (mitCTRL) zu betätigen, um das Tonsignal einzugeben, außerdem erinnert die Taste 'G' an den 1. Buchstaben von Glocke, die das Zeichen auch darstellen soll.

Noch ein weiteres Beispiel, wie man solche Steuerzeichen einsetzen kann: Eine Zahl soll eingegeben werden und soll in der Eingabezeile ohne Eingabetext erscheinen. Wir können programmieren:

```
10 cls:locate 1,10
20 INPUT "Zahl eingeben";n
30 locate 1,10:PRINT "            "n
```

oder die Zeile 30 anders, mit Steuerzeichen:

```
30 PRINT"KR"n
```

Das erste Steuerzeichen läßt den Text-Cursor wieder in die Eingabezeile zurückspringen. Das zweite Zeichen löscht diese Zeile.

Inverse Darstellung auf dem Bildschirm ist sehr einfach zu erreichen. Es wird nur das entsprechende Steuerzeichen vor und hinter den Text gestellt:

```
10 PRINT"XINVERSX"
```

Den gleichen Befehl können wir benutzen, um den gesamten Bildschirm invers darzustellen, das kann nützlich sein, wenn man Texte bearbeiten will und diese, wie auf Papier, dunkel auf hellem Hintergrund erscheinen sollen. Wir geben dazu ein im Direktmodus:

PRINT "X":CLS

Das Zeichen 4 steht für MODE. Wir können so programmieren:

```
10PRINT "D2klein"
20 FOR i=1 TO 1000:NEXT
10 PRINT "D0gross"
```

Zu den Steuerzeichen 8 bis 12 gibt es nicht viel zu sagen. Ein

Anwendungsbeispiel für Rücksprung in eine Zeile darüber hatten wir weiter oben bereits gesehen. Das Steuerzeichen 9 (Taste I und auch TAB) ist für den Bildschirm nicht interessant, weil man dafür ebensogut auch ein Leerzeichen setzen kann. Es wird erst interessant in Verbindung mit dem Drucker.
Wie schon gesagt, kann das Steuerzeichen 13 nur in der Form mit CHR$(13) eingesetzt werden.
Sehr interessant sind wieder die Steuerzeichen 16 bis 20, weil mit ihnen Teile einer Zeile oder auch ganze Bildschirmteile davor oder dahinter durch einen einzigen Befehl gelöscht werden können.
Nun noch ein Beispiel für Transparentmodus, Steuerzeichen 22. Zum Einschalten wird dem Steuerzeichen eine 1 angefügt, zum Ausschalten eine 0. Die Programmzeile dazu:

```
PRINT "Text-UnterstreichungHHHHHHHHHHHHHHHV1"
STRING$(15,"_")"V0."
```

Steuerzeichen 8 (Taste H) bringt den Textcursor zurück zum Anfang des Wortes „Unterstreichung". Dann wird im Transparentmodus der Unterstrich hinzugefügt und am Ende der Transparentmodus ausgeschaltet.
Das Zeichen 30 entspricht LOCATE 1,1.
Das Zeichen 31 ist wieder interessant, denn es entspricht dem Befehl LOCATE, wobei x und y in Form von weiteren Steuerzeichen angefügt werden. Das letzte Beispiel können wir mit diesen Zeichen wiederholen, es würde dann so aussehen:

```
10 PRINT "D2L0^JText-Unterstreichung0 # JV1"
STRING$(15,"-")"V0."
```

Um etwas klarer zu sehen, wollen wir die Steuerbefehle dieser Zeile analysieren: D2 steht für MODE 2. L steht für CLS. Die drei folgenden Zeichen gehören zusammen. Das erste Zeichen davon ist das Steuerzeichen der Taste Null, das für LOCATE steht. Das folgende Steuerzeichen steht für x, bzw. für die Spalte und hat den Wert 30, also Spalte 30. Wir haben hier den Fall, daß die Steuerzeichen im Zusammenhang mit dem vorstehenden Zeichen nicht ihre ursprüngliche Bedeutung haben, sondern als Parameter dazu, nun einen Zahlenwert darstellen. Nach dem folgenden Text kommt wieder eine LOCATE-Anweisung. Weil wir hier den Wert für Spalte 35 einsetzen müssen, aber keine Zahl einsetzen können, müssen wir ein Zeichen einsetzen, das den ASC-Code 35 hat. Es ist in diesem Falle das Zeichen "#", das als Parameter einzusetzen ist. Für Zeile 10 setzen wir wieder das Steuerzeichen mit diesem Wert ein. Danach folgt die Anweisung für den Transparentmodus mit dem Zeichen (Wert) 22, gefolgt von einer 1, wodurch die Einschaltung erfolgt. Dann kommt die Anweisung für den Unterstrich und zum Abschluß wieder das Zeichen für den Transparentmodus, aber mit einer Null, um diesen Modus auszuschalten. Das sieht vielleicht komplizierter aus, man kann sich aber daran gewöhnen. Nur ausdrucken kann man so ein Listing nicht mehr!
Es wäre interessant zu sehen, welcher Unterschied besteht zum normalen Programmieren ohne Steuerzeichen. Machen wir die Probe:

```
10 MODE 2:CLS:LOCATE 30,10
20 PRINT "Text-Unterstreichung"
30 LOCATE 35,10
40 PRINT CHR$(10)CHR$(1)STRING$(15,"-")CHR$(10)
CHR$(1)
```

Der Unterschied ist, daß wir für das Programm 127 Zeichen, Leerzeichen eingeschlossen, benötigt haben statt 61, wie in der ersten Version. Wenn Speicherplatz gespart werden muß, können die Steuerzeichen dazu beitragen. Manche Steuerzeichen sind einfach anzuwenden, andere wieder etwas schwieriger. Es sollte hier auch nur darauf hingewiesen werden, wie man mit solchen Kurzbefehlen umgehen kann, ohne auf alle Einzelheiten einzugehen. Wer daran interessiert ist, kann auch mit den Steuerzeichen für Farben experimentieren.
In der nächsten Ausgabe werden wir ausführlich auf die Befehle zur Druckersteuerung eingehen. Harold Eggink



Die Fehlermeldungen des BASIC 1.0 sind im Handbuch nur kurz erklärt – für Einsteiger oft zu kurz. Dieser Artikel gibt Ihnen Tips, wie Sie Fehler beheben können.

Unexpected NEXT
Die Erklärung im Handbuch verleitet manchen Anfänger dazu, zu glauben, der Computer könne jedes fehlende NEXT finden. Dem ist mitnichten so! Diese Meldung wird aber z. B. ausgegeben, wenn Sie eine Schleife mit FOR i=... eröffnen und mit NEXT j zu schließen versuchen. Hauptursache sind hier Tippfehler oder einfach Flüchtigkeitsfehler, wie bei folgendem Beispiel: FOR i=1 TO 5 : FOR j=1 TO 6 : ... : NEXT i : NEXT j
Darum merke: Bei Verschachtelungen von Schleifen stets die inneren zuerst schließen! Siehe auch 'NEXT missing'.

Syntax Error
Wer kennt ihn nicht? Meistens liegt einfach ein Tippfehler vor, der aber oft nur schwer zu finden ist. Hier hilft nur suchen!

Unexpected RETURN
Die häufigste Ursache hierfür ist ein vergessenes END. Da liegt z. B. das Hauptprogramm in den Zeilen 10 bis 500, und ein Unterprogramm steht ab 1000. Das Hauptprogramm läuft problemlos, und es ruft auch perfekt die Subroutine auf; aber nachdem Zeile 500 erreicht wurde, wird das UP nochmal ausgeführt, weil Sie vergessen haben, END einzutippen. Dieser Durchlauf durch das UP wurde jedoch nicht durch ein GOSUB verursacht. Folge: siehe oben!
Gefährlich ist es auch, Unterprogramme mitten ins Hauptprogramm zu setzen, um dann zu versuchen, sie mit GOTO zu umgehen. Wenn Sie das zu Beginn der Programmentwicklung tun, nehmen Sie später garantiert irgendeine Änderung vor, die Sie versehentlich mitten in irgendeinem 'Unter'programm landen läßt. Außerdem ist so ein Verfahren nicht gerade übersichtlich zu nennen. Wenn Sie sauber zwischen Hauptprogramm und Unterprogramm trennen, kann Ihnen ein 'Unexpected RETURN' schon nicht mehr zu schaffen machen.

DATA exhausted
Wahrscheinlich haben Sie eine DATA-Zeile vergessen oder unvollständig eingetippt. Wenn Sie die Daten innerhalb einer FOR-NEXT-Schleife lesen, kann es auch sein, daß die Anzahl der Wiederholungen oder die Schrittweite fehlerhaft sind.

Improper argument
Beispielsweise SQR(−1) oder LOG(0). Wenn die Meldung durch einen Befehl wie z=SQR(x) verursacht wurde: Überprüfen Sie, welchen Wert x hat. Höchstwahrscheinlich ist dieser Wert unzulässig, d.h. bei der Quadratwurzel: negativ. Verfolgen Sie von der angegebenen Zeile aus den Programmablauf zurück, wobei Sie nur die Zeilen betrachten, in denen x seinen Wert ändern kann. Sobald Sie festgestellt haben, wo x negativ werden kann, verhindern Sie dies durch eine Anweisung wie IF x < 0 THEN PRINT 'Wurzel nicht möglich.':END oder einfach x=ABS(x).
Übrigens: Es ist immer besser, sich gegen solche Fälle bereits vor der Eingabe des Programms durch gründliche Überlegung abzusichern.

Overflow
Da kann man nur raten: Rechnen Sie mit kleineren Zahlen!

Memory full
Versuchen Sie, Ihr Programm zu kürzen: Kommentare streichen, Textausgaben auf das Notwendigste beschränken. Auch IF-Abfragen können Sie oft durch Kürzeres ersetzen.

```
320 IF a=0 THEN GOTO 330 ELSE GOTO 340
330 ...
340 ...
```

läßt sich verkürzen auf:

```
320 IF a THEN 340
```

Monstren wie:

```
10 IF x=1 THEN 1000
20 IF x=2 THEN 2000
.
.
.
90 IF x=9 THEN 9000
```

können durch ON...GOTO ersetzt werden:

```
10 ON x GOTO 1000,2000,...,9000
```

Wenn Sie viele Matrizen definiert haben, sollten Sie auch prüfen, ob Sie z. B. bei DIM x (1000) auch wirklich alle Elemente bis 1000 benötigen oder nur 987. Das 'nullte' Element wird oft nicht verwendet, obwohl es auch Speicherplatz benötigt. Bei mehrdimensionalen Matrizen macht sich das ganz schön bemerkbar!
Scheuen Sie nicht davor zurück, DEFINT a-z zu verwenden, wenn Sie mit ganzen Zahlen auskommen. Integer-Zahlen benötigen im allgemeinen weniger Speicherplatz als reelle.

Line does not exist
Meistens tritt dieser Fehler dann auf, wenn Sie während der Programmentwicklung eine Zeile löschen, in die aber (z. B. durch ein IF) verzweigt wird.

Subscript out of range
Wenn Sie eine Matrix mit, sagen wir, DIM m(20) dimensioniert haben, können Sie logischerweise m(21) nicht mehr verwenden. Passen Sie evtl. die DIM-Anweisung an. Hüten Sie sich aber vor Ausdrücken wie m(x)! Überprüfen Sie hier zunächst den Wert von x. Wenn er außerhalb des Bereichs liegt, den Sie für die Matrix vorgesehen hatten, finden Sie (wie bei 'Improper Argument' beschrieben) heraus, woran das liegt, und verhindern Sie es. Negative Indizes wie m(-8) sind natürlich von Haus aus verboten; 0 hingegen ist erlaubt und sogar empfehlenswert – siehe 'Memory full'.
Dieser Fehler tritt auch auf, wenn die Anzahl der Indizes falsch ist, d.h. wenn Sie etwa DIM m(5,5) verwenden und später 'm(3)' sehen möchten.

Array already dimensioned
Beispiel:

```
100 DIM m(10)
.
.
.
995 DIM m(3)
```

Hier tritt der Fehler auf.
Die Abhilfe ist denkbar einfach: Gewöhnen Sie sich an, alle verwendeten Matrizen in einer einzigen DIM-Anweisung, möglichst am Programmanfang, zu dimensio-

nieren. Das verhindert diesen Fehler und erhöht die Übersichtlichkeit eines Listings enorm.
Trotzdem: gegen Fehler wie

```
100 DIM m(10),a(15),zz(2,8)
.
.
.
9000 IF Antwort$="J" THEN 100
```

ist niemand gefeit. Da hier 100 als fehlerhafte Zeile ausgegeben wird, können Sie u. U. lange nach der 'bösen' Zeile 9000 suchen. Ein Patentrezept gegen solche Art Selbstüberlistung gibt es leider nicht.

Division by zero
3/0 ist nicht definiert, und der CPC kann damit ebenfalls nichts anfangen. Bei Operationen wie z=5/x ist wieder der Wert von x zu kontrollieren. Sehr 'beliebt' ist auch hier eine Methode der Selbstüberlisting: In dem Bestreben, Formeln möglichst kompakt zu machen, setzt man Klammern – nur meistens einige zu wenig oder zu viel. Wenn Sie den Verdacht haben, daß dies die Ursache für die Fehlermeldung sein könnte, arbeiten Sie lieber mit Zwischenergebnissen.

Invalid direct command
INPUT x ist im Direktmodus blanker Unsinn. Diesen und ähnliche Befehle können Sie nur in einem Programm verwenden, sprich: nur mit vorangestellter Zeilennummer eingeben.

Type mismatch
Bei INPUT oder READ haben Sie hier wohl (z. B.) a statt a$ geschrieben – ein Tippfehler. Beachten Sie: INPUT a$ akzeptiert sowohl 13 als auch XXX als Eingabe, INPUT a logischerweise nur die 13. Bei x=x+a$ erhalten Sie dann aber spätestens die 'Type mismatch'-Meldung. Vor allem bei READ kann es vorkommen, daß Sie zwar eine Variable des richtigen Typs angegeben haben, aber in den DATAs etwas Unpassendes steht, z. B. hier:

```
10 FOR i=1 TO 2 : READ a$ : NEXT i
20 FOR i=1 TO 3 : READ a : NEXT i
30 DATA aaa,bbb,ccc,1,2,3
```

Ursache ist der Tippfehler '2' statt '3' in Zeile 10; als fehlerhafte Zeile wird aber 20 angegeben! (Das gleiche gilt für INPUT #9 und Dateien auf Cassette oder Diskette.)

String space full
Verwenden Sie weniger Strings – eine andere (einfache) Lösung gibt es nicht. Natürlich können Sie die eingegebenen Zeichen bzw. deren ASCII-Werte auch mit POKE irgendwo im Speicher ablegen, dann sind Sie nicht von den Strings im BASIC abhängig...

String too long
Hier sagt das Handbuch genug aus. Schwieriger wird es, wenn diese Meldung unterbleibt: Mit PRINT #9, STRING$ (200,"*");STRING$(200,"+") können Sie ohne weiteres ein Gebilde in eine Datei schreiben, das wie ein 'überlanger' String aussieht. Bei INPUT #9,a$ erhalten Sie jedoch nur die ersten 255 Zeichen davon als a$! Der Rest kommt erst mit dem nächsten INPUT; und das kann ganz schön Verwirrung stiften.

Unknown user function
Wenn Sie ein Listing abtippen, haben Sie die Zeile mit dem entsprechenden DEF FN-Kommando wahrscheinlich übersehen. Bei eigenen Programmen haben Sie sie vielleicht aus Versehen gelöscht.

RESUME missing
Schließen Sie Fehlerbehandlungsroutinen stets mit RESUME ab, wenn Sie sie schon verwenden müssen. Vor ON ERROR GOTO sei jedoch gewarnt! Sie können damit Fehler abfangen, die Sie überhaupt nicht abfangen wollten, und sich

so durch die 'Verschleierung' fehlerhafter Zeilen wieder einmal selbst austricksen. Versuchen Sie stattdessen, Zustände, die einen Fehler herbeiführen, durch geschickte Abfragen- und Variablenmanipulationen zu umgehen.

Unexpected RESUME
Hier gilt im Prinzip dasselbe wie für "Unexpected RETURN": Meist ist nur ein vergessenes END schuld an der Misere. Natürlich gehören auch Routinen zur Fehlerbehandlung nicht mitten ins Hauptprogramm!

Direct command found
Kann beim Abspeichern und Laden Ihrer selbstgeschriebenen Programme eigentlich nicht vorkommen, wenn Sie es nicht bei SAVEn gewaltsam provozieren. Da dies nur recht umständlich durchzuführen ist, wird Ihnen das kaum versehentlich passieren. Sollten Sie doch diese Fehlermeldung erhalten, liegt das wahrscheinlich an der Cassette oder Ihr Datenrecorder ist defekt.

Operand missing
INK 1 ist z. B. eine solch unvollständige Anweisung. Wahrscheinlich haben Sie den zweiten (und evtl. dritten) Operanden nur aus Flüchtigkeit vergessen.

Line too long
Das kann Ihnen zwar normalerweise auch nicht passieren; falls aber dennoch: einfach die eine zu lange Zeile auf zwei Zeilen mit erlaubter Länge verteilen.

EOF met
Wahrscheinlich versuchen Sie mehr aus einer Datei zu lesen, als darin enthalten ist. Beispielsweise so:

```
10 OPENOUT "test":FOR i=1 TO 10:PRINT #9,i:NEXT i:CLOSEOUT
20 OPENIN "test":FOR i=1 TO 11:INPUT #9,z:NEXT i:CLOSEIN
```

Wo nur zehn Zahlen gespeichert sind, können Sie schwerlich elf einlesen. Kontrollieren Sie also zunächst immer, ob auch soviele Eintragungen vorhanden sind, wie Sie einzulesen versuchen.
Wenn das der Fall ist, müssen Sie noch überprüfen, ob Sie beim Schreiben der Datei auch nicht das CLOSEOUT vergessen haben. Das kann Ihnen bis zu 2K an Daten einfach 'unterschlagen'.
Last not least könnte auch ein Bandfehler die Ursache sein. Versuchen Sie, die Datei auf einer anderen Cassette neu zu schreiben.

NEXT missing
Glauben Sie auch hier nicht, daß der Computer Ihnen alle fehlenden NEXTs aufspürt. Wenn aber eine Schleife bei Programmende immer noch nicht abgeschlossen wurde oder in Fällen wie der folgende, merkt er es schon:

```
10 FOR i=1 TO 8
.
.
.
95 FOR i=1 TO 2...
```

Der CPC würde hier korrekterweise Zeile 10 als falsch anzeigen und verrät Ihnen somit bereits das FOR, dem das NEXT fehlt. Andere Computer vermuten den Fehler hingegen in 95, da er sich erst dort bemerkbar macht.

File already open
Zweimal OPENIN oder OPENOUT hintereinander, ohne ein CLOSEIN oder CLOSEOUT durchgeführt zu haben, ist unzulässig. Diese Meldung kann Ihnen auch verraten, daß Sie irgendwo ein CLOSE... vergessen haben.

WEND missing/unexpected WEND
Im Prinzip gilt hier dasselbe wie für 'NEXT missing' und 'unexpected NEXT', nur auf WHILE-WEND-Schleifen übertragen.

Abschließend bliebe noch zu sagen, daß das Thema 'Fehlermeldungen' hier nur in der gebotenen Kürze behandelt werden konnte. Es ist naturgemäß unmöglich, auf alle möglichen Fehlerquellen einzugehen. Ich glaube jedoch, daß die bei Einsteigern häufigsten Ursachen aufgeführt wurden, und daß Ihnen damit ein Stück 'praktische Lebenshilfe' für die Computerei zur Verfügung steht. Lassen Sie sich von Fehlern nicht entmutigen – Programme, die auf Anhieb fehlerfrei laufen, gibt es nicht.
In diesem Sinne: Frisch ans Werk!

(Alwin Ertl)

Vergrößerte Zeichen

Das nachfolgende Programm vergrößert Buchstaben auf eine beliebige Größe. Wenn Sie das Programm „Screen Dump" aus Heft Nr. 5, Seite 49 vorher laden, dann können Sie die Buchstaben auch ausdrucken, indem Sie die Taste COPY betätigen. Wird die Taste CLR gedrückt, dann kann die Größe der Buchstaben geändert werden. Alle Cursor-Tasten funktionieren, Sie sehen den Cursor aber nicht. Die ENTER-Taste funktioniert auch. Der Cursor kann also in die nächste Zeile versetzt werden. Daß die Taste DEL funktioniert, ist selbstverständlich.

```
170 'Fuer
180 ' Schneider international programmie
rt
190 ' von:
200 ' Thomas Mischke (12 Jahre alt)
210 ' Bachgasse 21
220 ' 6074 Roedermark/Urberach
230 ' Tel. 06074/5293
240 '
250 ' Initialisierung
260 MODE 1:INK 1,24:INK 2,0:INK 0,0:PAPE
R 0:PEN 1
270 BORDER 0
280 ' Abfragen
290 INPUT"Wieviel mal in x-Richtung verg
roessern";xr
300 INPUT"Wieviel mal in y-Richtung verg
roessern";yr
310 h=400-yr*16
320 CLS:PEN 2
330 a$=INKEY$:IF a$="" THEN 330
340 IF a$=CHR$(241) THEN h=h-yr*16-yr:GO
TO 330
350 IF a$=CHR$(240) THEN h=h+yr*16+yr:GO
TO 330
360 IF a$=CHR$(243) THEN r=r+xr*16:IF r>
640-xr*16 THEN r=0:h=h-yr*16-yr:GOTO 330
  ELSE 330
370 IF a$=CHR$(242) THEN r=r-xr*16:IF r<
0 THEN r=640-xr*16:h=h+yr*16+yr:GOTO 330
  ELSE 330
380 IF a$=CHR$(13)  THEN r=0:h=h-yr*16-y
r:GOTO 330
390 IF a$=CHR$(&E0) THEN LOCATE 1,25:PRI
NT"
   ";:CALL &AA00:GOTO 330
400 IF a$=CHR$(&7F) THEN GOTO 10000
410 IF a$=CHR$(&10) THEN GOTO 20000
420 LOCATE 1,25:PRINT a$
430 FOR x%=0 TO 16 STEP 0.5
440 FOR y%=0 TO 16 STEP 0.5
450 IF TEST(x%,y%) THEN PLOT x%*xr+r,y%*
yr+h,1 ELSE PLOT x%*xr+r,y%*yr+h,0
460 NEXT:NEXT
470 r=r+xr*16:IF r>640-yr*16 THEN r=0:h=
h-yr*16-yr
480 GOTO 330
10000 LOCATE 1,25:PRINT" "
10005 r=r-xr*16
10010 FOR x%=0 TO 16:FOR y%=0 TO 16
```

```
10020 IF TEST(x%,y%) THEN PLOT x%*xr+r,y
%*yr+h,1 ELSE PLOT x%*xr+r,y%*yr+h,0
10030 NEXT:NEXT
10050 GOTO 330
20000 PEN 1:LOCATE 1,25:INPUT"Neue x-Ric
htung";xr
20010 LOCATE 1,25:INPUT"Neue y-Richtung"
;yr:PEN 2:GOTO 330
```

RSX Find

Der Find-Befehl sucht im Basic-Programm nach einer beliebigen Zeichenfolge, die vorher in eine Stringvariable geschrieben worden sein muß. Die Zeilen, in denen der Suchstring vorkommt, werden aufgelistet. Die gesuchten Zeichen müssen genauso übergeben werden, wie sie der List-Befehl auf dem Bildschirm darstellt. Basic-Befehlsworte müssen also groß geschrieben werden, falls man nicht nach Syntax-Fehlern sucht.
Das genaue Format des Befehls ist:

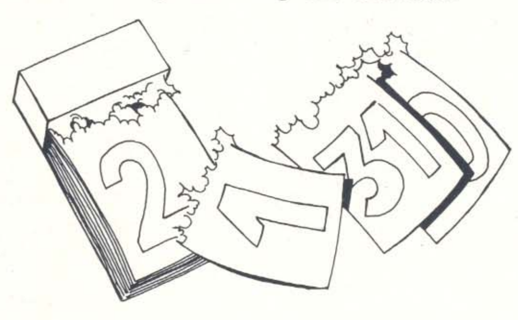
| FIND(,<erste Zeile>,<letzte Zeile>),<String-descriptoradresse>
Beispiel: I find$="NEXT":find,§find$ sucht im ganzen Programm nach NEXT.
| find,20,1000,§find$ sucht zwischen den Zeilen 20 und 1000 inklusive.
| find,30,-1,§find$ sucht ab Zeile 30 bis zum Ende.
| find,100,1000,§find$ listet Zeile 100 auf jeden Fall, wenn | find$=" " ist, da hinter jeder Zeilennummer ein Blank steht.

Da der Befehl nicht in den Direktmodus springt, ist es so möglich, eine oder mehrere Zeilen zu listen und im Programm fortzufahren, was mit List nicht möglich ist. Überhaupt läßt sich der Find-Befehl auch mit Variablen benutzen, was für den Suchstring nützlich sein kann.
Beispiel: for i=&61 to &66:a$="&"+chr$(i):I find,§a$:next.
Hiermit kann man nach klein geschriebenen Hex-Ziffer in Data-Zeilen suchen.
Mit dem Basic-Lader (FIND) läßt sich jeder beliebige Speicherbereich oberhalb von &7FFF wählen. Dazu muß nur zu Beginn des High-Byte der Startadresse (xx) in hexadezimaler Schreibweise angegeben werden. Das Programm belegt 255 Bytes auf den Adressen xx01 - xxFF, HIMEM wird auf xx00 gesetzt.
Das Programm läßt sich daher auch mit Floppy benutzen.

```
10 REM RSX BEFEHL FIND
20 REM DIE ZEICHEN IN <> MUESSEN MIT CTR
L EINGEGEBEN WERDEN
30 REM
40 CLOSEOUT
50 INPUT"<d>B<^>High-Byte der Startadres
se zweistellig hexadezimal";a$:IF LEN(a$
)<>2 OR VAL("&"+a$)<&80 THEN 50
60 a$=UPPER$(a$):PRINT"<Oac>Das Programm
 belegt die Speicher "a$"01 bis "a$"FF."
:a$="&"+a$:xx=VAL(a$):b$=a$+"00":start=V
AL(b$):b$=a$+"01":rsxint=VAL(b$):b$=a$+"
1A":find=VAL(b$)
70 MEMORY start:s=rsxint:RESTORE 120:GOS
UB 80:s=find:RESTORE 130:GOSUB 80:CALL r
sxint:END
80 z=0
90 READ x:IF x>255 THEN RETURN
100 IF x=&99 THEN x=xx
110 POKE s+z,x:z=z+1:GOTO 90
120 DATA &1,&C,&99,&21,&16,&99,&C3,&D1,&
BC,&0,&0,&11,&99,&C3,&1A,&99,&46,&49,&4E
,&C4,&0,&FFF
130 DATA &B7,&C8,&DD,&66,&1,&DD,&6E,&0,&
11,&5,&99,&1,&3,&0,&ED,&B0,&FE,&1,&28,&2
7,&FE,&3,&C0,&3A,&5,&99,&B7,&C8,&DD,&23,
&DD,&23,&DD,&22,&1,&99,&11,&1,&99,&2A,&1
,&99,&1,&4,&0,&ED,&B0
140 DATA &2A,&1,&99,&ED,&5B,&3,&99,&B7,&
ED,&52,&30,&B,&21,&FF,&FF,&22,&1,&99,&23
,&23,&22,&3,&99,&2A,&81,&AE,&23,&22,&8,&
99,&4E,&CD,&F8,&99,&C8,&46,&23,&5E,&23,&
56,&2A,&3,&99,&37,&ED,&52,&38,&9,&2A,&8,
&99,&9,&22,&8,&99,&18,&E5
150 DATA &CD,&0,&B9,&ED,&43,&0A,&99,&2A,
&8,&99,&CD,&63,&E1,&CD,&3C,&C4,&C5,&E1,&
B7,&ED,&5B,&6,&99,&1A,&11,&A4,&AC,&ED,&5
2,&E5,&C1,&EB,&ED,&B1,&28,&20,&2A,&8,&99
,&ED,&5B,&A,&99,&19,&CD,&F8,&99
160 DATA &C8,&2B,&22,&8,&99,&4E,&23,&46,
&23,&5E,&23,&56,&2A,&1,&99,&B7,&ED,&52,&
30,&C0,&C9,&E5,&C5,&F5,&ED,&4B,&4,&99,&5
,&28,&C,&ED,&5B,&6,&99,&13,&1A,&BE,&23,&
20,&2,&10,&F8,&D1,&7A,&C1,&E1,&28,&06,&C
B,&53,&20,&BC,&18,&BE
170 DATA &21,&A4,&AC,&CD,&45,&E1,&23,&7E
,&B7,&20,&F8,&CD,&4E,&C3,&18,&AE,&3E,&00
,&86,&23,&86,&FE,&0,&C9,&FFF
```

Kalender

Nach der Eingabe des Datums sagt Ihnen der CPC sofort, auf welchen Wochentag dieser Tag fällt bzw. fiel!!



```
10 REM ewiger Kalender
20 DIM t$(7)
30 t$(1)= "SONNTAG"
40 t$(2)= "MONTAG"
50 t$(3)= "DIENSTAG"
60 t$(4)= "MITTWOCH"
70 t$(5)= "DONNERSTAG"
80 t$(6)= "FREITAG"
90 t$(7)= "SAMSTAG"
100 PRINT " Datum (TAG, MONAT, JAHR):"
110 INPUT D, M, J
120 IF J> 1752 THEN 150
130 PRINT "Kein Datum vor 1753 !!"
140 GOTO 100
150 K= INT(0.6 + (1/m))
160 l = j-k
170 o= m+12 *k
180 p=l/100
190 z1= INT(p/4)
200 z2=INT(p)
210 z3= INT((5*l)/4)
220 z4= INT(13*(o+1)/5)
230 z = z4+z3-z2+z1+d-1
240 z=z - (7*INT(z/7))+1
250 PRINT "der "; d;".";m;".";j;
260 PRINT" ist ein ";
270 PRINT t$(z)
280 PRINT
290 GOTO 100
```

Einführung in das Betriebssystem CP/M 2.2

Wie komfortabel ein Betriebssystem sein kann, wird schnell bei einer Verkettung von mehreren Betriebssystembefehlen deutlich. Mit dem Befehl SUBMIT können solche Folgen von Befehlen einfach zusammengefaßt werden.

Submit

Submit erlaubt die sequenzielle Abarbeitung vom CP/M-Kommando, ohne zusätzlichen Benutzereingriff. Um den Befehl Submit zu gebrauchen, müssen Sie zuvor eine Datei des Typs ".SUB" erstellen. Diese Datei kann sowohl vom CP/M-Kontex-Editor als auch von einem Textverarbeitungsprogramm erzeugt werden. Die Datei sollte die Liste der CP/M-Kommandos in der Reihenfolge enthalten, in der sie ausgeführt werden sollen, und zwar jeweils eins pro Zeile.

Submit kann zum Beispiel nach jeder Programmausführung mit den Programmen DISCCOPY, COPYDISK, FILECOPY etc. verbunden werden. Es können zum Beispiel nach jedem LOGO-Aufruf Dateien des Typs "*.LOG" auf Laufwerk B: kopiert werden. Zuerst sollte man das Kommando PIP.COM von der A- auf die B-Seite kopieren. Geben Sie bitte folgenden Befehl im Kommando-Mode von CP/M und der Systemdiskette im Laufwerk A: ein:

A > FILECOPY PIP.COM

Der Aufruf wird durch folgende Meldung bestätigt:

FILECOPY V2.1
Please insert SOURCE disc into drive A then press any Key:_

Nach dieser Meldung müssen Sie nur Enter drücken. Nach anschließender, kurzer Ladezeit erscheint eine neue Meldung:

Copying startet...
Please insert DESTINATION disc into drive A then press any Key:_

Hierauf sollten Sie die Diskette umdrehen. Achten Sie darauf, daß kein Schreibschutz besteht und benutzen Sie nur eine kopierte Systemdiskette. Das Original sollte sicher aufgehoben und nicht zu irgendwelchen Tests benutzt werden (der Schreibschutz einer 3"-Diskette ist **aktiv,** wenn der weiße Schieber eingefahren ist). Haben Sie alles erforderliche getan, dann müssen Sie nur noch die Enter-Taste drücken.

Folgende Meldung bezeugt dann noch einmal das korrekte Ende von FILECOPY:

PIP > .COM Copiet.
Copying complete
Please insert a CP/M system disc into drive A then press any Key:_

Nach erneutem Drücken der Enter-Taste meldet sich FILECOPY ab:

FILECOPY V2.1 finished
A >

Jetzt haben wir PIP.COM von der A-auf die B-Seite kopiert. Als nächstes muß ein File des Typs ".SUB" erstellt werden. Dazu können wir das PIP-Kommando verwenden:

A> PIP COPYLOG.SUB=CON:

Der Cursor wird nun eine Zeile tiefer gehen, und Sie können den ersten Befehl: **LOGO** schreiben. Ein Zeilensprung muß mit I M (=CONTROL-M) erzeugt werden. Ist I J ausgeführt worden, so kann man die zweite Zeile, wie folgt beschreiben:

PIP B:=*.LOG

Um nun zum Kommando-Mode von CP/M zurückzukehren, ist es notwendig, die Zeile mit Control-Z abzuschließen; ein Control-C (Warmstart) funktioniert nicht. Sie erhalten darauf, unmittelbar nach dem Schließen der Datei, daß PROMPT-Zeichen mit der Angabe des zur Zeit primären Laufwerkes.

Wenn dies nun alles funktioniert hat, so können wir probieren, ob die neugeschaffene Befehlsfolge richtig arbeitet. Geben Sie nun folgendes Kommando ein:

A>SUBMIT COPYLOG

Daraufhin müßte, wie von Geisterhand eingetippt, auf dem Bildschirm die Aufforderungsmeldung mit einem nachfolgendem LOGO erscheinen und der LOGO Interpreter wird sich mit der üblichen Bereitschaftsmeldung zu erkennen geben. Falls Sie sich noch nie mit LOGO beschäftigt haben, so geben Sie einfach die folgenden Befehle ein:

```
to quat                          [ENTER]
cs                               [ENTER]
repeat 4 [fd 100 rt 90]          [ENTER]
end                              [ENTER]
```

Manche von Ihnen werden sich fragen, was dieses Programm tut. Sie können es selbst ausprobieren. Tippen Sie einfach noch einmal das Wort "quat" ein. Zuerst wird der Bildschirm gelöscht (cs). Anschließend wird die Befehlsfolge 100 Punkte nach vorne und dann um 90 Grad drehend 4 mal wiederholt. Daraus entsteht natürlich ein Quadrat.

Unser eigentliches Problem war die Abspeicherung eines LOGO-Programms mit einer darauffolgenden Rückkehr zum CP/M. Folgende Befehle sind einzutippen:

```
save "viereck                    [ENTER]
bye                              [ENTER]
```

Normalerweise, wenn LOGO durch CP/M gestartet wurde, müßte sich an dieser Stelle das Aufforderungskommando zeigen und man könnte einfach weiter arbeiten. Da LOGO aber durch SUBMIT gestartet wurde, werden zuerst die in der .SUB-DATEI noch fehlenden Kommandos abgearbeitet. Es folgt also ein PIP-Aufruf mit dem Ziel das erstellte Programm oder die erstellten Programme (nämlich die des Typs .LOG) vom Laufwerk A: auf das Laufwerk B: zu kopieren. Nun sind wir wieder im CP/M-Mode. Doch SUBMIT kann noch einiges mehr:

Während man in BASIC an die Benutzung von Variablen gewöhnt ist, muß man in der Betriebssystemebene häufig darauf verzichten. Nicht so bei CP/M 2.0 oder genauer bei SUBMIT. Doch leider bleibt die Anzahl der benutzten Variablen auf 9 beschränkt. Sie werden durch ein beginnendes $ gekennzeichnet und enden mit einer Nummer von 1 - 9. Folgende Syntax kann also auch benutzt werden:

SUBMIT name A B C D E F G H I

Die Ziffern A - I betiteln dabei irgendeinen Namen, der nachher bei der Submit-Datei in die Variablen $1, $2 usw. eingesetzt werden. Der erste dem SUBMIT folgende Name wird immer $1. Der darauffolgende, zweite Name $2 usw. Es wird freigestellt, wieviele der 9 Variablen benutzt werden.

Beispiel:

Wollen Sie eine Datei auf Laufwerk A editieren, nach Beendigung auf Laufwerk B: speichern und anschließend noch die Kapazität der Datei vom Laufwerk B: prüfen, so empfiehlt sich SUBMIT zu benutzten. Folgendes muß dazu gewandt werden:

1. SUB-File erstellen
2. SUBMIT aufrufen

Die Realisierung könnte dazu etwa so aussehen:

```
A > PIP EDSAVE.SUB=CON:
ED $1.$2
PIP B:=A:$1.$2[V]
STAT B:$1.$2
```

Wie bereits bekannt sein dürfte, beendet man die Eingabe mit einem Control-Z. Probieren Sie aus, ob es funktioniert. Wir wenden uns jetzt einem weiteren Leckerbissen von CP/M zu: dem Befehl XSUB. XSUB wurde mit dem Ziel konzipiert, Tastatureingaben in die Kommando-Datei des Tpys .SUB mit einzutragen. Dabei ist es notwendig, daß vor einer Tastaturabfrage der Befehl XSUB ausgeführt wurde. XSUB kann nur in einer SUBMIT-Datei erscheinen und bleibt bis zu einem Kaltstart im Speicher. XSUB lädt sich selbst in den Bereich des CCP (Console Command Processor) und meldet sich von dort aus mit "xsub active" nach jedem Warmstart und bei einem Aufruf.

Es ist beispielsweise denkbar, folgende Zeilen in eine .SUB-Datei zu schreiben:

```
XSUB
DDT
I$1.HEX
R
GO
SAVE 2 $2.COM
```

Speichern wir ein File mit solchem Inhalt unter dem Namen SICHERE.SUB ab und rufen dann mit

SUBMIT SICHERE VERSUCH1 VERSUCH 2

dieses File auf, so wird das FILE VERSUCH1.HEX mit dem Namen VERSUCH2.COM nach einer Ausführung gespeichert.

Doch genug der langen Worte, jetzt wird es Zeit, daß Sie alles am Bildschirm ausprobieren. Keine Angst, es hört sich viel schwieriger an, als es eigentlich ist. Wenn Sie alles verstanden haben und Ihre eigenen SUBMIT-Dateien erstellen, dann wissen Sie jetzt bestimmt, wozu Betriebssysteme nützlich sind. Im nächsten Teil werden wir uns mit dem CP/M-Editor ED.COM noch einmal näher befassen. Sollten Sie jedoch weitere Fragen haben, steht Ihnen unser „Heiße Draht" oder ein formloses Schreiben zur Verfügung. (HF)

Wir suchen:

Freiberufliche Mitarbeiter zur Ergänzung und Unterstützung unseres Redaktions-Teams. Wenn Sie sich mit dem CPC 464/664 auskennen, Programmierung bzw. Hardware keine Fremdworte sind, dann sollten Sie sich auf jeden Fall mit uns in Verbindung setzen.

Wir erwarten:
- fundierte Kenntnisse auf dem Gebiet der Programmierung bzw. der Hardware
- einen flotten Schreibstil
- Kreativität und sorgfältiges, dabei selbständiges Arbeiten.

Wir bieten:
- langfristige Zusammenarbeit
- der Leistung entsprechende Honorare
- die Möglichkeit, in einem aufgeschlossenen, jungen Team mitzuarbeiten, auch wenn Sie bisher keine einschlägigen Erfahrungen bzw. Tätigkeiten auf diesem Gebiet besitzen.

Auf Ihre schriftliche Bewerbung freuen wir uns schon jetzt. Wir antworten umgehend!
Ihre Unterlagen senden Sie bitte an:
Data Media GmbH
Schneider CPC International
3440 Eschwege, Fuldaer Str. 6

Wir sind:
ein junger, aufstrebender Zeitschriftenverlag, der mit seinem Produkt zur Zeit konkurrenzlos auf dem Markt etabliert ist. Unsere Aktivitäten erweitern sich um ein Vielfaches, daher müssen wir expandieren.

Biete an Software

★ Van der Zalm - Software ★
präsentiert TOP-Anwendungen
z. B. FIBU, FAKTU etc. Liste
bei Fa. Elfriede van der Zalm,
Schieferstätte, 2949 Wangerland 3.

CPC-464 50 Progr. auf Cassette
gegen DM 40,- in Scheinen an
M. Günsche, Postfach 5604,
8700 Würzburg 1. Keine Raubkopien!
Kein Telefon vorhanden.

★ **und weitere CPC-Software...** ★
★ Ausführl. INFO g. 1.30 DM ★
★ Software-Service Doussier ★
★ Carl-Miele-Str. 179 ★
★ 4830 Gütersloh 1 ★
★ Händleranfragen erwünscht ★

★★ CPC 464 + 664-Software ★★
★ Floppy-Manager nur 49.00 ★
★ Textverarbeitung 98.00 ★
★ Dateiverwaltung 69.00 ★
★ Adressenverwaltung 69.00 ★
★ Lagerverwaltung 89.00 ★
★ Fakturierung 69.00 ★
★ Vokabel-Manager 49.00 ★
★ und weitere CPC-Software... ★
★ Ausführl. Info g. 1.30 DM ★
★ **Software-Service Doussier** ★
★ Carl-Miele-Str. 179 ★
★ 4830 Gütersloh 1 ★
★★ Händleranfragen erwünscht ★★

★**Kniffel**★=32k/nur Vorkasse o.NN
Liste DM10/Kass.DM20/Disk.DM30
Fickinger, Finken 21, 8901 Diedorf

Qualitätssoftware Fa. W. Bögemann
23 Quarnbeh, Kirschgarten 13
T.:04340/8212 auch Sa+So ab 14.00
Pgr. für Geschäft-Büro-Hobby-
Unterricht z.B. Journalbuchführung
vom Finanzamt geprüft. 420,-
Tgl. Vorführung + Beratung auf
CPC464-Commod.PC10-Sharp-Oric

Viele preiswerte Spiele, Mathe,
Anwend. (z.B. MINIVISICALC) von
2-25 DM! Katalog geg. 1 DM von
SCHNEIDERSOFT Andreas Wagner,
Gartenstr. 4, 8201 Neubeuern

Für Schneider CPC
Wärmedarf nach DIN 4701, FI-BU
Lohnbuchh., Kassenbuch
Lichttechn. Berechnungen
El.-Leitungsberechnungen
Information anfordern:
Kleinmann, Im Defdahl 51
4600 Dortmund ☎**0231/516199**

Originalprogramme, Anwend. + Spiele
zum 1/2 Preis + Porto
J. Seidler, 28 Bremen,
Bremerhavenerstr.96(0421/3961493)

Psychologische Tests
3 Tests auf CPC 464/Persönlich-
keitsprofil u.a./51,- N.N.
Bene Piut, 030/78801161,
030/6221880, 030/4342098.

CPC 464 FORTH83
mit Turtlegrafik, Tracer,
Editor, dt. Handbuch, Cassette
148,– DM Info:
FORTH-SYSTEME Angelika Flesch,
Pf. 1226, 7820 Titisee-Neustadt,
Tel. 07651/1665

Lohn- und Gehaltsabrechnung
mit allen gesetzlichen Vorschriften
für 464/664 sofort lieferbar.
Komplett auf 3"-Diskette
DM 445,-
STS GmbH
Albertstr. 40, 6052 Mühlheim (M)
Tel. 06108/75828

Super-Graph.Adventure 'Die Jubeltraube' f. CPC 464/664 68 Räume
25 Graph.-Absp.Spielstand-Eing.
Ganz. Sätze in Deutsch. Disc 35DM
Cass 25DM-Scheck o. bar beilegen.
Bitte angeb. Disc/Cass/Farbe/Grün
an A. Persch, Ackerfeldstr. 20
4200 Oberhausen 11

Sharp-Laufwerke am Schneider
DDI-1 – 4 x 40 Tracks IBM 34
8" Disk. gebr. a 5,- DM
Horst Fritz, 04344/2431 ab 18 Uhr

★ **CPC 464 Englisch lernen!!!**
★ Das Programm enthält viele
★ Vokabeln und über 100 unre-
★ gelmäßige Verben. Auf Kassette
★ nur 14,- DM. Sie können zu-
★ sätzliche Vokabeln eingeben
★ und abspeichern. T. Schray,
★Altenbachweg 8/1,7120 Bietigheim.

DISKPEEK, direct disk editor-
für Schneider CPC 464 - Anzeige und
Änderung von Sektoren auf der Floppy
(Diskette- bitte Vortex- oder
Schneider-Format angeben), DM 45,-
M. Stiemke, Leinenweberweg 7,
4 Düsseldorf 13, T.:0211/7182741.

Je 1x verk. ich Org. z. 1/2 Preis
Budget-Manager Disk 1x Easy-Top-
Calc 1x The Quill, 1x Jonny Reb
1x Amsgolf, 1x Football Manager,
1x Mini-Office, 1x Ghostbusters,
1xDecathlon, 1x Masterchess, 1x American Football Org. Verp. zus. 150,- DM
Tel. 07472/24271 ab 19.30.

★★★**Graphic Programmer**★★★
Erstellen Sie Graphiken zum
Einbau in eigene Programme Joy.
od. Tast. Draw, Circle, Line, Save,
Text, Load... DM 21,- + 6,- DM Porto
Bar oder Scheck an
Rene Sion,
Walther-Rathenaustr.16, 5 Köln 50.

Anwendungssoftware N. C. Warmbold
Falkenweg 23, 2110 Buchholz

Für die Schule! Notenverwaltung
berechnet Schnitt, Halbj.+Zeugnis-
noten nach eig. Wichtung. Klassen-
+ Notenlisten, Notenspiegel, An-
zeige Schirm/Druck NLQ401 o.ä.
Cass. Unk. 50,- DM inkl. Porto, Info
Freiumschl. C. Bernhold, Rommelstr. 31,
8783 Hammelburg 2.

Profess. Astrologieprg.: 5 Seit.
Persönlk.Analyse+Grafik+Aspekt.
Schumacher, Carl-Zeiss-Str. 14
3200 Hildesheim ☎**05121/23646**

50 Programme für CPC 464 nur
40,- DM (keine Raubkopien), Vorkasse
erbeten. F. Herwig, Am Bostelberg 1c,
3170 Gifhorn, kein Tel.

UNI-DAT 3.1 - die universelle Datei
für DM 29,- ★ Disc. DM 39,-
gleich bestellen bei: F. Worms,
Weidegrund 25, 4780 Lippstadt

★★ **Lotto-Neuheit für CPC 464** ★★
Einsetzen des CPC für Systeme u. Festlegen Wahlzahlen.
★ Spielen mit 10 - 24 Wahlzahlen,
gewinnen ab 3 richt. Wahlzahlen.
★ Neues Verfahren zum Festlegen,
d.h. finden der Wahlzahlen.
★ Einsatz ab DM 20,-, Geld-zurück-
Garantie auf alle Systeme.
★ Wahlzahlen eingeben, System
erstellt Tippreihen für Sie!
★ Gewinnzahlen eingeben, richtige
pro Reihe werden angezeigt.
★ Spezialauswertungen (ausgeblie-
bene Zehnergruppen, Verteilung
gerade/ungerade, häufigste als 1.
bis 6. gezogene Zahlen.
★ Ziehungshäufigkeit aller Zahlen
seit Beginn des Lottos enthalten.
★ Rückstand der Zahlen absolut
und in Wochen wird angezeigt.
★ Die Auswertungen werden in
Zahlen und grafisch dargestellt.
★ Spezialverfahren (Ausschluß-
verfahren) zum Finden der
Wahlzahlen.
★ Alle Einzelziehungen der
letzten fünf Jahre integriert
im System.
★ Updaten jede Woche durch
Eingabe der aktuellen Gewinn-
zahlen.
Dieses Programm hat 70 KB;
Preis DM 88,-. Info DM 0,80,
Democassette DM 5,- (wird bei
Kauf voll angerechnet). Discette
+ DM 10,-.
Wetzel Software, Steinstraße 22,
6843 Biblis 2.

Biete an Hardware

Orig. Schneider Floppy
Laufwerk mit Controller, gebraucht,
1a Zustand, DM 500,-,
M. Stiemke, Leinenweberweg 7,
4 Düsseldorf 13, T.:0211/7182741

CPC Hardware und Software
Günstig und schnell
M + H Computerhandel GbR
Eisenacher Weg 4, 4000 Düsseldorf 12

Der Pac-Assembler (neu) DM 90,-.
Cumara 5 1/4" Floppy m. Utility-Disk.
DM 450,-/Kl. Bauer,Wastl-Witt-Str.56,
8000 München 21.

Verkaufe CPC 464 Gruenmonitor,
4 Spiele, Joystick, Firmware
Handbuch, noch 7 Mon. Garantie
VB 1.050,- DM Tel. 06106-24646

Verkaufe: Schneider CPC 464 Farbe
Floppy DDI und FD-1 sowie
Drucker NLQ 401 gegen Gebot.
Außerdem Data-Becker und Heim-
Verlag-Bücher sowie Software
40% unter Neupreis.
Tel. 02171-89232 ab 18.00.

CPC 464 Colour, 1/2 Jahr alt + DDI-1
zu verkaufen. DM 1.700,- VB.
W. Slawik, Monheim T.:02173/62211.

Leistungsfähige Diskettenstation für den Schneider CPC 464
5 1/4" 708 KB, 80 Spuren, mit
Controller, CP/M 2.2 Systemdiskette.
1 Laufw. 1198.00 DM; 2 Laufw.
1698.00 DM keine Hard- oder
Software-Anpassungen nötig.
Huber Elektronik, Wörnitzstr. 3,
8850 Donauwörth Tel. 0906/8743.

Sprachsynthesizer dk-tronics,
DM 89,-, Tel. 040/6472427 OK!

Antireflexscheibe für CPC 464
vermindert lästige Reflexe auf
dem Bildschirm. DM 22,-.
Angabe Mono- oder Farbmonitor.
D, Kamp, Köllmannstr. 13,
4300 Essen 14, Tel. 0201/594323.

■■■ Schneider CPC 464 ■■■
Serielle Schnittstelle-RS232
mit Software, Drucker
Expansionsport weiter nutzbar,
199,– DM, S. Schrader, 2805 Stuhr 5,
T.: 04221/52268 o. 04206/9123 Gw

CPC 464: univers. 32-Bit-TTL-
Ein-/Ausgabe-Interface, inter-
ruptfähig, mit Netzteil DM 198,–
Dazu: **EPROMER** 2716-27128, 2516-
2564 incl. Betriebssyst. DM 179,–
M&L Electronic, Hauptstr. 1 a,
5511 Mannebach, T.: 06581/3535

Suche Software

Suche Zimmerei-Abbundprogr. CPC
Tel. 08652/7883.

Suche Kopierprogramm (f. Sicher-
heitskopien), Ass/Dias., gute
Grafikprgrm. u.a. gute Programme
H. Thür,
Am Brautsee 46, 2380 Schleswig.

Software gesucht! Intern. Ver-
marktung Ihrer Software! Wir
suchen: Spiele in MC, Adventure,
Utilities und anderes. Wenden
Sie sich an uns, evtl. sogar
feste Mitarbeit möglich.
Firma Gebr. Eckhardt, Computer-
Software, Tel. 02135/50952!

Suche Programme für den CPC 464 und C 64
G. Pasquero, Beurhausstraße 5
4600 Dortmund

Verschiedenes

★★★ Staubschutzhauben ★★★
CPC 464/664, Floppy DM 15,-. Monitor sw/color DM 35,- Fa. Weber, Pf. 201237, 56 Wuppertal.

Drucke Ihre Listings kostenlos
– auch in NLQ – aus. Disc./Kassette mit ausreichend Rückporto an:
H. Thür,
Am Brautsee 46, 2380 Schleswig.

Lösungshilfen für alle Adventures gesucht. Ebert, Mainstraße 51, 4100 Duisburg 1.

Suche 'Schneider CPC International' Heft 3 + 4 Tel. 02150/1897.

Suche Kontakt zu anderen CPC-Usern in München Tel. 93 69 75.

Suche CPC-Kontakte Baum MK/UN Tel. 02373/76247 ab 16.00 Uhr.

Suche Hardware

Suche Floppy Tel.04131/128888.

Das ist Ihre Chance... schon eine Kleinanzeige bringt oftmals großen Erfolg und hilft neue Kontakte knüpfen. Nutzen Sie unser Angebot und profitieren Sie von der Tatsache, daß unsere Zeitschrift „Schneider CPC International" jeden Monat von mehreren zigtausend Computer-Interessierten gelesen wird. Möchten Sie etwas verkaufen, tauschen oder suchen Sie das „Tüpfelchen auf dem „i" – dann sollten Sie die eigens hierfür bestimmte Bestellkarte im Heft ausfüllen und an unseren Verlag absenden. Ihre Annonce erscheint dann in der nächsterreichbaren Ausgabe.

Wir möchten ausdrücklich darauf hinweisen, daß wir keine Anzeigen veröffentlichen, aus denen ersichtlich ist, daß es sich hierbei um Veräußerungen von Raubkopien oder dergleichen handelt.
Die Redaktion

Achtung! An alle CPC-User-Clubs! Suchen Sie noch Mitglieder oder wollen Kontakte zu anderen CPC-Usern oder Clubs knüpfen? Dann schreiben Sie uns doch einfach. Wir veröffentlichen jede Anschrift eines CPC-User-Cubs gratis! Falls auch Ihr User-Club sich einmal vorstellen möchte (evt. mit Foto), nutzen Sie unsere Zeitschrift als Verbindungsglied und Kontaktadresse!

'Schneider-Gesellen'
zwecks Clubgründung im Kreis Kleve und Krefeld gesucht.
Thomas Lavis, Mühlenstr. 12, 4182 Uedem, Tel.:02825/8665.

CPC-Clubgründung angestrebt
für die Bereiche Gütersloh Wiedenbrück, Harsewinkel, Verl, Reiner Lotte, 4830 Gütersloh, Tel. 05241/36350.

'CPC-Clubs in Wien gesucht!'
SINN Alexander, Postfach 219, 1140 Wien.

CPC-Userclub Freiburg/Br.
c/o Ulrich C. Manns, Lehenerstr. 146, 7800 Freiburg/Br., Tel.:(0761)891448.

Schneider CPC-Club
Reutlingen-Alb und Umgebung. Info darüber bei Rüdiger Wörner, An Rainen 22, 7420 Münsingen.

CPC-C sucht Mitglieder.
Info: Philip Matthey, 5, Rue des Sapins, Senningerberg, Luxembourg.

CPC User-Club:
Sebastian Mandl, Anderlingstr. 23, 7063 Welzheim, Tel.:07182/2192.

CPC-Profisoft-User-Club,
c/o Andreas Dauer, Hagenbeckstr. 106, 2000 Hamburg 54, T.:040/5401424.

CPC-User-CLUB East Belgium,
Kirchplatz 28, B-4720 Klemis/Belgien.

CPC Clubs:
Christian Rainer, Planoetzenhofstr. 1
6020 Innsbruck.
und
Markus Haslwanter, An-der-Lanstr. 18, 6020 Innsbruck.

Jürgen Geiger, Weinbergstr. 18, 7129 Talheim., Tel.:07133/6799.

Hans Denk
Gartenstraße 14
8531 Dietersheim

Christian Rainer
Planötzenhofstraße 1
A-6020 Innsbruck

Maluche Lothar
Peter-Dörfler-Straße 70
8998 Lindenberg/Allgäu
Tel.: 08381/7468

CPC-Club
Gerd Engelbarth
Aschener Weg 1
4503 Dissen a.T.W

Computer-Club "Tapferes Schneiderlein"
Hans-Jürgen Lugauer
Frankenstraße 83
8700 Würzburg
oder
Jan Agne
Florian-Geyer-Straße 14
8700 Würzburg
Info-Austausch mit anderen Clubs erwünscht!

CAUC Gütersloh
Matthias Hovestadt
Kirchstraße 25
4836 Herzebroch-Clarholz
Tel.: 05245/6120

Schneider CPC-User-Club
Wolfgang A. Jaeger
Postfach 1731
7600 Offenburg

Computer-Club im Bezirk Niederrhein
Bernd Beeking
Im Mühlenfeld 15
4193 Kranenburg
Telefon: 02826/5470

8900-User-Augsburg
suchen CPC-User zwecks Info Austausch. Anschrift:
Thomas Wipper
Gärtnerstraße 17
8900 Augsburg

SCC Sons of the Light
Bernhard Kübler
Haunstetter Straße 22
8900 Augsburg
Tel.: 0821/573873

CPC-User Club
Kirchweg 30
6300 Giessen

Schneider Computerclub
Roger Ritter
Domagkweg 7
3000 Hannover 61
Info gegen –,80 DM Rückporto

Gründungsinitiative des Schneider CPC-464 Professional User-Club Saar
Infos bei Herbert Weingärtner
Tel.: 06894-83 42

Schneider Computer-Club Heinsberg/Rheinland
Helmut Krings
Josef Gasperstraße 25
D-5138 Heinsberg 1
Tel.: 02452/5710



10/85

erhalten Sie ab 25. September bei Ihrem Zeitschriftenhändler

Berichte

Textverarbeitung auf den CPC's
- große Marktübersicht und Vergleichstests von Textverarbeitungsprogrammen!

CPC 6128
- unsere Erfahrungen mit dem neuesten Computer der Schneider-Familie wollen wir Ihnen nicht vorenthalten.

Serie

Entwicklung der Mikroelektronik
In unserer geschichtlichen Rundreise wenden wir uns diesmal den Entwicklungen der Computertechnik zu. Dazu zählen Schlagworte wie Silicon Valley, Chip, Transistor etc.

Programme

Superda
- Arcade-Game der Extraklasse! Versäumen Sie nicht dieses aufregende Weltraum-Abenteuer.

Universelle Dateiverwaltung
- die Profisoftware

Monitor
- Software für Maschinenspracheprogrammierer

Lehrgänge

Lehrgänge
- Z-80-Assembler
- Grafik-Kurs
- Sound mit dem CPC
- Basic für Einsteiger
- CP/M in der Anwendung

Software Reviews

‚C' für den CPC
DevPac 80
Alien – das Spiel zum Film
Subterranean Stryker
Pinball Wizzard

Tips und Tricks

- Sprites!
- wertvolle Hilfen zum NLQ-Drucker!
- Extended Basic (Befehlserweiterung)

Hardware

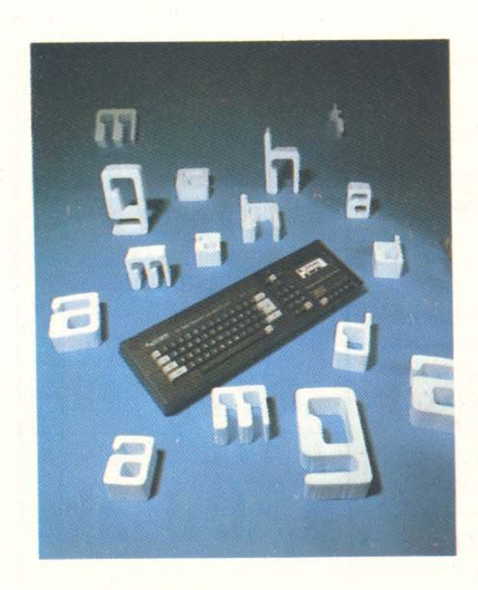
Drucker auf dem Prüfstand
- wieder haben wir mehrere Matrixdrucker für Sie getestet und unsere Ergebnisse in einem großen Testbericht festgehalten.

Rund um den CPC

Aktuelles aus der Branche und Kurzinfos zu den neuesten Schneider-Produkten.



Drucker auf dem CPC-Prüfstand. Wieder haben wir mehrere Drucker für Sie getestet und unsere Ergebnisse in einem großen Testbericht festgehalten.



Textverarbeitung
Auf dem Markt tummeln sich zur Zeit verschiedene Textverarbeitungsprogramme.
Worauf beim Kauf zu achten ist und wie leistungsfähig jedes einzelne von ihnen ist, erfahren Sie in unserem großen Testbericht.

Inserentenverzeichnis

Ausverkauft!

Es erreichen uns noch immer Zuschriften, in denen um Nachlieferung der Ausgaben 3 und 4 von Schneider CPC International gebeten wird. Diese Ausgaben sind restlos vergriffen und daher nicht mehr lieferbar!

Übrigens war das März-Heft (3/85) die allererste Ausgabe; die Hefte 1 und 2/85 gibt es daher nicht. Von den Ausgaben 5/85, 6/85 und 7/85 stehen noch Restposten zur Verfügung. Wir bitten, dies bei evtl. Nachbestellungen zu berücksichtigen.

Spiele-Software

Titel		Best.-Nr.	Preis
Gems of Stradus	(Amsoft)	1215	39,– DM
Laserwarp	(Amsoft)	1216	39,– DM
Haunted Hedges	(Amsoft)	1217	39,– DM
Galactic Plague	(Amsoft)	1223	39,– DM
Splatt	(Amsoft)	1224	39,– DM
Jammin	(Amsoft)	1226	39,– DM
Fruit Machine	(Amsoft	1228	39,– DM
Mutant Monty	(Amsoft)	1229	39,– DM
Loopy Laundry	(Amsoft)	1230	39,– DM
Blagger	(Amsoft)	1231	39,– DM
Classic Racing	(Amsoft)	1232	39,– DM
Crazy Golf	(Amsoft)	1233	39,– DM
Roland on the run	(Amsoft)	1234	39,– DM
Detective (Cludo)	(Amsoft)	1235	39,– DM
Hunter Killer	(Amsoft)	1237	39,– DM
Jet Boot Jack	(Amsoft)	1238	39,– DM
Ghouls	(Micro Power)	1242	29,– DM
The Prize	(Amsoft)	1243	39,– DM
Sultans Maze	(Amsoft)	1245	39,– DM
Oh Mummy	(Amsoft)	1247	39,– DM
3D Invaders	(Amsoft)	1249	39,– DM
Alien Break in	(Romik)	1250	27,– DM
Atom Smasher	(Romik)	1251	27,– DM
3D Monster Chase	(Romik)	1252	27,– DM
Dragons Gold	(Romik)	1253	27,– DM
Star Commando	(Terminal)	1255	27,– DM
Electro Freddy	(Amsoft)	1257	39,– DM
Punchy	(Amsoft)	1259	39,– DM
Stockmarket	(Amsoft)	1260	39,– DM
Bridge it	(Amsoft)	1261	39,– DM
Cubit	(Amsoft)	1262	39,– DM
Grand Prix Drive	(Amsoft)	1263	39,– DM
Killer Gorilla	(Micro Power)	1270	49,– DM
Sorcery	(Virgin)	1273	29,– DM
Map Rallye	(Amsoft)	1239	34,– DM
Traffic	(Amsoft)	1241	34,– DM
Xanagrams	(Amsoft)	1256	34,– DM
Admiral Graf Spee	(Amsoft)	1264	34,– DM
Roland goes digging	(Amsoft)	1265	34,– DM
Space Hawks	(Amsoft)	1267	34,– DM
Quack a Jack	(Amsoft)	1268	34,– DM
Classic Adventure	(Amsoft)	1271	34,– DM
American Football	(Amsoft)	1275	34,– DM
Frank-N-Stein	(Amsoft)	1276	34,– DM
Super Pipeline II	(Amsoft)	1277	34,– DM
The Scouts steps out	(Amsoft)	1278	34,– DM
Stud Poker/Stud Jack	(Data Media)	1211	39,– DM
Disk 3"		1411	49,– DM
Disk 5 1/4"		1611	49,– DM
Skat/Bauernskat	(Data Media)	1212	39,– DM
Disk 3"		1412	49,– DM
Disk 5 1/4"		1612	49,– DM
Poker/17+4	(Data Media)	1213	39,– DM
Disk 3"		1413	49,– DM
Disk 5 1/4"		1613	49,– DM
Zargon	(Data Media)	1214	39,– DM
Disk 3"		1414	49,– DM
Disk 5 1/4"		1614	49,– DM
Codename Mat	(Amsoft)	1218	39,– DM
Disk 3"		1418	49,– DM
Snooker	(Amsoft)	1220	39,– DM
Disk 3"		1420	49,– DM
Roland ahoy	(Amsoft)	1222	39,– DM
Disk 3"		1422	49,– DM
Pyjamarama	(Amsoft)	1236	39,– DM
Disk 3"		1436	49,– DM
Spannerman	(Amsoft)	1246	39,– DM
Disk 3"		1446	49,– DM
Masterchess	(Microgen)	1248	39,– DM
Disk 3"		1448	49,– DM
Roland in time	(Amsoft)	1266	39,– DM
Disk 3"		1466	49,– DM

Titel		Best.-Nr.	Preis
ER*Bert	(Microbyte)	1272	39,– DM
Disk 3"		1472	49,– DM
Disk 5 1/4"		1672	49,– DM
Future World	(Data Media)	1289	39,– DM
Disk 3"		1489	49,– DM
Disk 5 1/4"		1689	49,– DM
Enterprise	(Data Media)	1294	49,– DM
Disk 3"		1494	59,– DM
Disk 5 1/4"		1694	59,– DM
Der Rote Baron	(Data Media)	1295	39,– DM
Disk 3"		1495	49,– DM
Disk 5 1/4"		1695	49,– DM
Das Geheimnis der 4 Juwelen	(Data Media)	1296	39,– DM
Disk 3"		1496	49,– DM
Disk 5 1/4"		1696	49,– DM
Pittmans Typing Tutor	(Amsoft)	1240	49,– DM
Disk 3"		1440	54,– DM
Harrier Attack	(Amsoft)	1244	39,– DM
Disk 3"		1444	49,– DM
Amsgolf	(Amsoft)	1269	34,– DM
Disk 3"		1469	44,– DM

Neu Marco Polo (Data Media)

Die Marco Polo Saga. Ein deutsches Grafik-Adventure.

Marco Polo 1	Disk 3"	1497	49,– DM
	Disk 5 1/4"	1697	49,– DM
Marco Polo 2	Disk 3"	1498	49,– DM
	Disk 5 1/4"	1698	49,– DM
Marco Polo 3	Disk 3"	1499	49,– DM
	Disk 5 1/4"	1699	49,– DM

Die Thorr-Trilogie (Data Media)

Eine der komplexesten deutschen Textadventure-Trilogien. Erleben Sie Thorr, die Welt der Zauberer, Ritter und der Schwarzen Magie.
Die Thorr-Adventures sind auch einzeln zu erhalten:

Die letzten Tage von Burg Ghorrodt	1290	39,– DM
Disk 3"	1490	49,– DM
Disk 5 1/4"	1690	49,– DM
Die Fluch nach Thyrros	1291	39,– DM
Disk 3"	1491	49,– DM
Disk 5 1/4"	1691	49,– DM
Das Geheimnis von Thyrros	1292	39,– DM
Disk 3"	1492	49,– DM
Disk 5 1/4"	1692	49,– DM
Die Thorr-Trilogie komplett	1293	99,– DM
Disk 3"	1493	109,– DM
Disk 5 1/4"	1693	109,– DM

Bücher

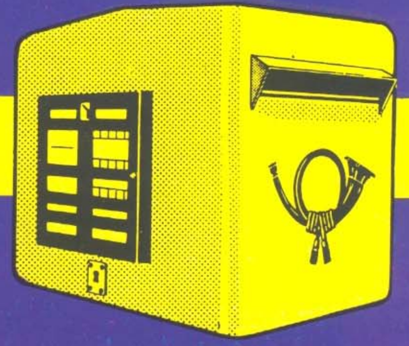
Huslik:

Inside out, 428 Seiten	**59,00 DM**

Data Becker:

Das Floppy-Buch zum CPC, ca. 250 Seiten	**49,00 DM**
CPC Hardware-Erweiterungen, ca. 300 Seiten	**49,00 DM**
Peeks & Pokes zum CPC, ca. 220 Seiten	**29,00 DM**
CPC 464 Graphik & Sound, 220 Seiten	**39,00 DM**
CPC 464 Intern, 548 Seiten	**69,00 DM**
Das Maschinensprachebuch zum CPC 464, 330 Seiten	**39,00 DM**
CPC Tips & Tricks, 263 Seiten	**39,00 DM**
Adventures – und wie man sie auf dem CPC 464 programmiert, 320 S.	**39,00 DM**
CPC 464 Basic-Programme, 185 Seiten	**39,00 DM**
Das Basic-Trainingsbuch zum CPC 464, 285 Seiten	**39,00 DM**
Das Schulbuch zum CPC 464, 389 Seiten	**49,00 DM**

Wie bestellen?
Die Bestellkarte im Innenteil der CPC heraustrennen, ausfüllen und an DATA MEDIA senden. Lieferung erfolgt umgehend! Zahlung per Vorkasse oder Nachnahme zzgl. Porto-bzw. Nachnahmegebühr. (Nachnahme ins Ausland ist nicht möglich.)



data media gmbh MAILORDER

Weitere Artikel in unserem Gesamtkatalog. Bitte anfordern (2,– DM für Rückporto beilegen.)
DATA MEDIA GmbH, Bereich Mailorder, Postfach 1263, 4620 Castrop-Rauxel
Telefonische Auskünfte: 0231/12 50 71
In der Schweiz: Data Media Schweiz, Bungertstraße 15, 8802 Zürich-Kilchberg, Tel.: (01) 715 36 19

Schneider
Bildschirmtext



Schneider TV made in Germany